TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ALEKSI KAURTO

ETHERNET-POHJAISTEN AUTOMAATIOVERKKOJEN REAALI-AIKAINEN KUNNONVALVONTA

Diplomityö

Tarkastajat: professori (emeritus)
Hannu Koivisto ja
professori Matti Vilkko
Tarkastajat ja aihe hyväksytty
28. helmikuuta 2018

TIIVISTELMÄ

ALEKSI KAURTO: Ethernet-pohjaisten automaatioverkkojen reaaliaikainen kunnonvalvonta
Tampereen teknillinen yliopisto
Diplomityö, 76 sivua, 4 liitesivua
Lokakuu 2018
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Automaation tietotekniikka
Tarkastajat: professori (emeritus) Hannu Koivisto ja professori Matti Vilkko

Avainsanat: verkonvalvonta, automaatioverkko, viitemalli, Ethernet, FCAPS

Automaatiojärjestelmien nopeasti kasvava tiedonsiirtotarve luo yhä suurempia haasteita automaation tietoverkoille sekä niiden ylläpidolle. Näiden haasteiden vähentämiseksi verkkoa tulee valvoa sekä hallita verkkoympäristön vaatimalla laajuudella. Tämän tutkimuksen painopiste on verkonvalvonnassa ja sen soveltamisessa automaatiojärjestelmien Ethernet-pohjaisiin tietoliikenneverkkoihin. Verkonvalvonnan tulee olla jokapäiväinen työkalu yrityksen verkkoympäristön toiminnan tukena. Valvontajärjestelmän avulla varmistetaan verkon oikea toiminta sekä vähennetään suunnittelemattomia tuotantoa häiritseviä katkoksia.

Tässä diplomityössä tutkittiin kirjallisuusselvityksen avulla verkonvalvonnan menetelmiä ja käytäntöjä perinteisen informaatioteknologian näkökulmasta. Työssä pohdittiin näiden menetelmien ja käytäntöjen soveltuvuutta Ethernet-pohjaisiin automaatioverkkoihin. Aluksi perehdyttiin automaatioverkkoihin ja näiden verkkojen vaatimuksiin. Vaatimusten pohjalta ymmärrettiin verkon toimintaan vaikuttavat tekijät sekä toiminnan varmistamisen kriittisyys. Tiedonsiirtoprotokollien standardiviitemallien avulla luotiin teoriapohja automaatioverkon profiileille sekä verkonvalvontaan soveltuville protokollille. Tutkimuksessa perehdyttiin myös virtausteknologioihin sekä näiden teknologioiden hyödyntämiseen automaation verkonvalvonnan tarkoituksissa. Verkonhallinnan viitemallit tarjoavat ohjeistuksen verkonhallinnan ja -valvonnan osa-alueiden toteuttamiseksi sekä liittämiseksi osaksi yrityksen jokapäiväistä liiketoimintaa.

Tämän diplomityön tavoitteena oli toteuttaa teoriapohjan perusteella verkonvalvontakokokonaisuus, joka soveltuu työn tilaajan tarpeisiin. Verkonvalvontakokonaisuus toteutettiin avoimen lähdekoodin sovelluksilla. Näiden sovellusten avulla vastattiin vikojenvalvonnan sekä suorituskyvynvalvonnan osa-alueiden vaatimuksiin. Verkonvalvontajärjestelmän avulla voidaan vähentää verkon vikaantumista ja lyhentää vianselvityksen kestoa. Valvontajärjestelmän avulla toteutettiin verkon rakenteen visualisointi sekä lisättiin verkon läpinäkyvyyttä verkonylläpitäjälle.

ABSTRACT

ALEKSI KAURTO: Real-time condition monitoring of Ethernet based automation networks
Tampere University of Technology
Master of Science Thesis, 76 pages, 4 Appendix pages
October 2018
Master's Degree Programme in Automation Engineering
Major: Information Systems in Automation
Examiners: Professor (emeritus) Hannu Koivisto and Professor Matti Vilkko

Keywords: network monitoring, industrial network, reference model, Ethernet, FCAPS

Fast growing need for information transfer in automation systems creates even bigger challenges for industrial networks and their maintenance. To avoid these challenges network needs to be monitored and managed as the extent of network infrastructure demands. Main focus of this research is on network monitoring and applying it to Ethernet based information networks in automation systems. Network monitoring should be a daily tool for enterprise's network infrastructure operations support. With the assist of monitoring system is assured networks correct function and reduce unplanned disturbing breaks in the production.

This thesis studied with the literature review, network monitoring methods and practices from the traditional informational technology perspective. This research debated how these methods and practices applies to Ethernet based automation networks. Automation networks and their demands were studied first. Based on these demands it was understood the influencing factors in network operation and the criticality of ensuring the operation. With a support of standard reference models of data transfer protocols, a theoretical base was created for automation network profiles and for protocols suited for network monitoring. This research also got acquainted with flow technologies and how these technologies can be utilized in automation network monitoring. Network management reference models offer a framework for fulfilling network management and monitoring fields, and implementing those as part of enterprise's business.

The thesis goal was to create, based on theoretical review, a network monitoring concept, which is applicable with the orderer's needs. The network monitoring concept was composed through an open source software. With these software answered for fault and performance monitoring fields requirements. It is possible to reduce network failures and to shorten the troubleshooting duration with the network monitoring system. With the monitoring system, the network structure visualization was implemented and increased the network transparency for the administrator.

ALKUSANAT

Ensiksi haluan kiittää työn tilaajaa diplomityöaiheesta sekä mahdollisuudesta tehdä diplomityö muiden työtehtävien ohella. Kiitos tilaajalle myös työn ohjaamisesta, kärsivällisyydestä sekä ajasta, jonka sain käyttää tutkimuksen tekemiseen.

Haluan kiittää motivoinnista sekä diplomityön kirjallisen osuuden ohjaamisesta ja tarkastamisesta professori (emeritus) Hannu Koivistoa sekä professori Matti Vilkkoa. Kiitos myös Jari Seppälälle hyvistä neuvoista sekä ajatuksista matkan varrella.

Kiitos vielä perheelle, joka on jaksanut kannustaa työn tekemisessä alusta loppuun sekä erityisesti Anulle, jonka tuen ansiosta olen saanut vietyä työni päätökseen.

Tampereella, 31.10.2018

Aleksi Kaurto

SISÄLLYSLUETTELO

KUVALUETTELO

TAULUKKOLUETTELO

LYHENTEET JA MERKINNÄT

AES	Advanced Encryption Standard, lohkosalausmenetelmä.
API	Application Programming Interface, ohjelmointirajapinta.
ARP	Address Resolution Protocol, protokolla verkkoyhteyskerroksen osoitteen muuntamiseksi siirtoyhteyskerroksen osoitteeksi.
ASN.1	Abstract Syntax Notation 1, rajapinnan kuvauskieli.
ATM	Asynchronous Transfer Mode, pakettikytkentäinen asynkroninen tiedonsriitotapa.
BASE	Baseband, Ethernet-teknologian peruskaistansiirto.
CIP	Common Industrial Protocol, EtherNet/IP:n, ControlNetin sekä DeviceNetin yhteinen sovelluskerroksen protokolla.
CP	Communication Profile, IEC 61784 standardin mukainen kommunikointiprofiili.
CPF	Communication Profile Family, IEC 61784 standardin mukainen kommunikointiprofiiliperhe.
CRC	Cyclic Redundancy Check, tarkisteavaimen luontiin tarkoitettu tiivistealgoritmi.
CSMA/CD	Carrier Sense Multiple Access with Collision Detection, menetelmä useiden laitteiden kesken jaetun verkon varaamiseksi ja törmäysten tunnistamiseksi.
DNS	Dynamic Name Service, nimipalvelujärjestelmä.
eTOM	enhanced Telecom Operations Map, teleoperaatioiden liiketoimintaprosessien viitekehys.
FAB	Fulfillment, Assurance, and Billing, verkonhallinnan toimintamalli.
FCAPS	Faults, Configuration, Accounting, Performance and Security, verkonhallinnan toimintamalli.
FCS	Frame Check Sequence, kommunikointiprotokollien kehyksissä käytetty virheentarkastus menetelmä.
FTP	File Transfer Protocol, protokolla tiedostojen siirtämiseen.
GSM	Global System for Mobile Communications, maailmanlaajuisesti käytetty matkapuhelinjärjestelmä.
HTTP	Hypertext Transfer Protocol, hypertekstin siirtoprotokolla.
IANA	Internet Assigned Numbers Authority, maailmanlaajuinen organisaatio, joka vastaa esimerkiksi IP-osoitteiden jakelemisesta.
ICMP	Internet Control Message Protocol, protokolla tiedonsiirron diagnostiikan lähettämiseksi.
IDS	Intrusion Detection System, tunkeutumisen tunnistusjärjestelmä.
IEC	International Electrotechnical Comission, kansainvälinen sähköalan standardointiorganisaatio.
IEEE	Institute of Electrical and Electronics Engineers, kansainvälinen tekniikan alan järjestö.
IETF	Internet Engineering Task Force, Internet-protokollien standardoinnista vastaava organisaatio.
IGMP	Internet Group Management Protocol, protokolla ryhmälähetystietojen hallintaan.
IP	Internet Protocol, Internet-protokolla.
IPFIX	IP Flow Export protocol, virtausteknologia verkkoliikenteen analysointiin.

ISO	International Organization for Standardization, kansainvälinen standardointijärjestö.
ITU-T	International Telecommunications Union – Telecommunication Standardization Sector, kansainvälisen televiestintäliiton televiestinnän standardointisektori.
JSON	JavaScript Object Notation, yksinkertainen avoin tiedostomuoto tiedonvälitykseen.
LAN	Local Area Network, lähiverkko.
MAC	Media Access Control, IEEE 802 standardin mukaisen verkon varauksesta ja liikennöinnistä huolehtiva protokolla.
MAN	Metropolitan Area Network, alueverkko.
MIB	Management Information Base, verkkolaitteen hallittavien objektien tietovarasto.
MRTG	Multi Router Traffic Grapher, sovellus, jonka avulla voidaan kerätä SNMP-objekteja ja tuottaa niistä graafisia kuvaajia.
OAM&P	Operations, Administration, Maintenance and Provisioning, verkonhallinnan toimintamalli.
ODVA	Open DeviceNet Vendors Association, organisaatio, joka vastaa CIP-protokollan ylläpidosta ja kehityksestä.
OID	Object Identifier, yksilöintitunnus.
OSI	Open Systems Interconnection, viitemalli avointen järjestelmien välisien yhteyksien määrittelemiseksi.
PAN	Personal Area Network, likiverkko.
PI	Profibus and Profinet International, Profibus ja Profinet teknologioiden kehityksestä vastaava yhteisö.
PING	Packet Internet Groper, työkalu TCP/IP verkkolaitteiden välisen yhteyden testaamiseen.
Profibus DP	Profibus Distributed Periphery, hajautettujen kenttälaitteiden väliseen sarjaliikenne kommunikointiin tarkoitettu kenttäväylä.
RFC	Request for Comments, IETF-organisaation julkaisemia Internet-asiakirjoja.
RRD	Round Robin Database, aikajatkuvan tiedon tallennukseen käytettävä tietovarasto.
RTPS	Real-Time Publisher Subscriber, Modbus/TCP-protokollan reaaliaikalaajennos.
SCTP	Steram Control Transport Protocol, tiedonsiirtoprotokolla.
sFlow	Sampled flow, näytteistykseen perustuva virtausteknologia.
SMI	Structure of Management Information, Määrittelee MIB-tietovarastoissa käytettävän hallintatietojen rakenteen ja syntaksin.
SMTP	Simple Mail Transfer Protocol, sähköpostiviestien välittämiseen tarkoitettu protokolla.
SNMP	Simple Network Management Protocol, TCP/IP-verkkojen hallintaprotokolla.
SSH	Secure Shell, salattuun tietoliikenteeseen tarkoitettu protokolla.
Syslog	System Logging Protocol, järjestelmäsanomien välittämiseen käytettävä protokolla.
TCP	Transmission Control Protocol, yhteydellinen tiedonsiirtoprotokolla.
TCP/IP	Transmission Control Protocol / Internet Protocol, kokoelma verkkojenväliseen kommunikointiin käytettävistä protokollista.

TMN	Telecommunication Management Network, tietoliikenneverkkojen hallinnan viitekehys.
TOM	Telecoms Operation Map, teleoperaatioiden elinkaarenhallinnan viitekehys.
UDP	User Datagram Protocol, yhteydetön tiedonsiirtoprotokolla.
UHF	Ultra High Frequency, radioliikenteessä käytettävä suurtaajuusalue.
WAN	Wide Area Network, laajaverkko.
WLAN	Wireless Local Area Network, langaton lähiverkko.
XDR	External Data Representation, sFlow-sanomien määrittelyssä käytettävä standardi.

1 JOHDANTO

Automaatiojärjestelmien jatkuvasti lisääntyvä tiedonsiirtotarve luo yhä suurempia haasteita automaation tietoverkoille ja niiden ylläpidolle. Ethernet-pohjaisissa automaatioverkoissa siirretään erilaisten protokollien avulla suuria määriä tietoa automaatiojärjestelmän tarpeita varten. Automaatiojärjestelmä ei välttämättä sijaitse kokonaisuudessaan fyysisesti samalla alueella, jolloin verkkojen rakenne monimutkaistuu. Laitosten välisien yhteyksien lukumäärän kasvaessa kymmeniin tai satoihin, muuttuu verkon rakenne hankalasti hallittavaksi sekä ylläpidettäväksi.

Automaatioverkon vikaantuessa ongelman selvittäminen vie paljon aikaa ja resursseja sekä aiheuttaa ylimääräisiä kustannuksia. Ongelman selvittäminen vaatia usein myös erityisosaamista, mitä ei yleensä löydy automaatiojärjestelmän haltijalta itseltään. Jos verkkoja valvotaan reaaliaikaisesti, ongelmat olisi mahdollista tunnistaa ja korjata jo ennen tuotantoa häiritseviä katkoksia. Reaaliaikaisen verkonvalvonnan avulla ongelmat pystytään tunnistamaan ja paikallistamaan välittömästi oikeaan sijaintiin sekä laitteeseen, jolloin korjaustoimenpiteet voidaan aloittaa välittömästi.

1.1 Tutkimuskysymykset ja -menetelmät

Diplomityön tarkoituksena on tutkia Ethernet-pohjaisten teollisuusautomaatioverkkojen kunnonvalvontaa sekä selvittää informaatioteknologiassa käytettyjen kunnonvalvontamenetelmien soveltuvuus työn tilaajan tarpeisiin. Työn tilaaja haluaa selvittää verkonvalvonnan menetelmiä ja käytäntöjä sekä niiden soveltuvuutta teollisuuden automaatioverkkoihin. Tutkimuksessa hyödynnetään avoimen lähdekoodin työkaluja, joiden avulla koostetaan automaatioverkkoon soveltuva kunnonvalvontakokonaisuus. Kunnonvalvontaan käytettävä työkalukokonaisuus halutaan pitää toimittaja riippumattomana ja kokonaisuuden tulee olla laajennettavissa sekä muokattavissa jatkuvasti muuttuvia tarpeita varten. Diplomityön tutkimuskysymykset ovat:

1. Mitkä ovat informaatioteknologiassa yleisesti hyväksytyt verkonhallinta ja -valvonta periaatteet sekä mitkä ovat niiden yleiset käytännöt?
2. Mitkä ovat verkonvalvonnan menetelmät ja kuinka ne soveltuvat teollisuusautomaation Ethernet-pohjaisiin tietoliikenneverkkoihin?
3. Soveltuvatko avoimen lähdekoodin työkalut yrityksen automaatioverkkovalvonnan tarpeisiin?

Kahteen ensimmäiseen tutkimuskysymykseen etsitään ratkaisua kirjallisuustutkimuksen avulla. Kolmanteen tutkimuskysymykseen vastataan toteutusosassa hyödyntäen kahden

ensimmäisen kysymyksen perusteella luotua teoriapohjaa. Toteutusosiossa koostetaan verkonvalvontakokonaisuus avoimen lähdekoodin sovelluksilla, joista pääosissa ovat Nagios Core sekä Elastic Stack. Nämä avoimen lähdekoodin sovellukset ovat valikoituneet diplomityön toteutusosioon asiakastarpeen sekä yrityksen aikaisemman tutkimuksen ja kokemuksen perusteella. Kyseisiä sovelluksia ei yrityksessä ole käytetty aikaisemmin verkonvalvontatarkoituksessa ja tutkimuksen avulla pyritään selvittämään niiden soveltuvuus kyseiseen tarkoitukseen. Edellä mainitut avoimen lähdekoodin alustat mahdollistavat yrityksen olemassa olevien sekä kehitteillä olevien ominaisuuksien integroinnin ja yhdistämisen yhdeksi kokonaisuudeksi.

Työhön on valittu kolme automaatioverkon tiedonsiirtoprotokollaa, jotka ovat EtherNet/IP, Profinet sekä Modbus/TCP. Kyseiset tiedonsiirtoprotokollat on valittu, koska Profinet sekä Modbus/TCP ovat tilaajan laajasti käyttämiä protokollia ja näiden lisäksi tilaaja suunnittelee EtherNet/IP-protokollan laajempaa käyttöönottoa tulevaisuudessa. Verkonvalvontaan soveltuvien protokollien valinta perustuu verkkolaitteiden valmistajien tukemiin protokolliin, joita voidaan hyödyntää myös automaatioverkoissa ja jotka voisivat soveltua automaatioverkon laitteiden kunnonvalvontaan.

1.2 Rakenne

Tämä tutkimus koostuu johdannon lisäksi seuraavista osista:

Luku 2 esittelee tutkimuksen kannalta oleellisia termejä sekä tarkastelee automaatioverkon ja kaupallisen verkon eroja.

Luku 3 esittelee tietoliikenneverkon kerrokset standardiviitemallien avulla. Lisäksi käydään läpi automaation Ethernet-pohjaisia tiedonsiirtoprotokollia sekä tarkastellaan niiden sijoittumista standardiviitemalleihin ja TCP/IP (Transmission Control Protocol / Internet Protocol) -protokollapinoon.

Luku 4 esittelee verkonhallinnan viitekehyksiä ja tarkastelee näiden avulla verkon kunnonvalvonnan vaatimuksia. Lisäksi perehdytään verkonvalvontaan soveltuviin protokolliin sekä virtausteknologioihin.

Luku 5 määrittelee aikaisempien lukujen teoriapohjan avulla verkonvalvontasovelluksen toteutuksen sekä esittelee toteutuksessa käytetyt avoimen lähdekoodin sovellukset ja tekniikat. Lisäksi pohditaan tutkimuksen aikana löydettyjä jatkotutkimus ja -kehityskohteita, jotka voidaan toteuttaa tulevaisuudessa.

Luku 6 sisältää lyhyen yhteenvedon työn tuloksista.

2 VERKOT

Tässä luvussa käydään läpi työn kannalta oleellisia termejä sekä selvitetään automaatioverkon ja kaupallisen verkon eroja. Alaluvussa 2.3 käydään läpi automaatioverkon ominaisuuksia sekä vaatimuksia.

Yleisesti termillä verkko voidaan viitata mihin tahansa toisiinsa yhdistettyihin ryhmiin tai järjestelmiin, jotka pystyvät jakamaan informaatiota keskenään toistensa välillä. Tietoliikennetekniikassa termillä verkko tarkoitetaan kahden tai useamman laitteen liittämistä yhteen tietoliikenneyhteyksien avulla. [1-3] Verkot voivat olla yhteydessä myös tosiin verkkoihin sekä sisältää aliverkkoja [1]. Tiedon siirtäminen lähteeltä määränpäähän voi tapahtua yksittäisen laitteen läpi, mutta yleensä tiedon kuljettamiseksi haluttuun määränpäähän tarvitaan useita laitteita. Tietoliikenneverkko on laitteiston ja ohjelmiston kokonaisuus, joka mahdollistaa käyttäjien välisen tiedonvaihdon. Tietoliikenneverkot ovat saaneet alkunsa tarpeesta jakaa tietoa oikea-aikaisesti. Tiedon jakaminen ja levittäminen verkkojen avulla ovat kriittisiä toimintoja jokaiselle nykyaikaiselle yritykselle. [2]

Verkot voidaan luokitella niiden maantieteellisen kattavuuden mukaan ja yleisiä luokitteluita ovat likiverkko, PAN (Personal Area Network), lähiverkko, LAN (Local Area Network), alueverkko, MAN (Metropolitan Area Network), sekä laajaverkko, WAN (Wide Area Network). Likiverkkoja käytetään tavallisesti noin 10 metrin etäisyydellä toisistaan olevien laitteiden yhdistämisessä. Likiverkkoon liitettävät laitteet ovat tyypillisesti matkapuhelimia, tabletteja sekä kannettavia tietokoneita. [2] Lähiverkolla tarkoitetaan maantieteellisesti pienen alueen tietoliikenneverkkoa, kuten yhden rakennuksen kattavaa verkkoa, jossa on suuri nopeus sekä siirtokapasiteetti. Lähiverkot ovat yleisesti yrityksien omassa hallinnassa, mutta ne on mahdollista tuottaa palveluina ulkopuolisen yrityksen toimesta. [2-4] Alueverkko on yleisnimitys verkoille, jotka kattavat kaupungin, kuntayhtymän, yliopiston tai taajama-alueen. Alueverkkoja käytetään yhdistämään erillään sijaitsevia lähiverkkoja toisiinsa. [2, 3] Laajaverkolla tarkoitetaan verkkoja, jotka ulottuvat paikkakunnalta toiselle tai maan rajojen ulkopuolelle. Nämä verkot ovat julkisten teleoperaattoreiden hallinnoimia verkkoja. [2, 3, 5]

2.1 Ethernet

IEEE (Institute of Electrical and Electronics Engineers) 802.3 CSMA/CD (Carrier Sense Multiple Access with Collision Detection) työryhmä on standardisoinut Ethernet-teknologioita useiden vuosien ajan [4, 6]. Ethernet on lähiverkkoteknologia, joka on langallisten lähiverkkojen markkinajohtaja [4]. Ethernet ei ole vain yksi standardoitu

protokolla, vaan useiden standardoitujen tekniikoiden, kuten 10BASE-T, 10BASE-2, 1000BASE-LX ja 10GBASE-T yleisnimitys. [4] Teknologiaa kuvaava kolmiosainen lyhenne koostuu teknologian tukemasta nopeudesta, käytettävästä signaalityypistä sekä fyysisen siirtotien tyypistä. Suurin osa työryhmän määrittelemistä tekniikoista perustuu peruskaistansiirtoon, BASE (Baseband), joka tarkoittaa, että fyysinen siirtotie huolehtii ainoastaan Ethernet-signaloinnin kuljetuksesta. Fyysisenä siirtotienä on alun perin käytetty koaksiaalikaapelia, mutta nykyisin siirtotiet toteutetaan kierretyllä parikaapelilla tai valokuidulla. [4, 6]

Ethernetin alkuperäinen toimitila perustuu CSMA/CD:n MAC (Media Access Control) -protokollaan. MAC-protokolla määrittelee säännöt kehysten lähettämiseksi jaettuun Ethernet-kanavaan, josta käytetään myös nimitystä half-duplex. [3, 6] Ethernetin toimitilasta, jossa kaksi laitetta voivat lähettää ja vastaanottaa kehyksiä samanaikaisesti, käytetään nimitystä Full-duplex. Tässä toimitilassa ei ole tarvetta CSMA/CD-algoritmille, koska väylää ei jaeta useiden laitteiden kesken eikä kehysten törmäyksiä tästä syystä synny. [4, 6]

Ethernetin tehtävä on huolehtia lähiverkon laitteiden välisestä tiedonsiirrosta [3, 6]. Ethernet kattaa OSI-viitemallin (Open Systems Interconnection Reference Model) kaksi ensimmäistä kerrosta, jotka ovat fyysinen kerros ja siirtoyhteyskerros [4]. OSI-viitemallia käsitellään tarkemmin alaluvussa 3.1.

2.2 Lähiverkkotopologiat

Teollisuuden verkot ovat tyypillisesti hajautettuja ja vaihtelevat ympäristöstä riippuen useilla osa-alueilla, kuten käytettävien protokollien ja verkon rakenteen eli topologian osalta. Yritysverkkojen tavoin teollisuuden viestintäjärjestelmät voidaan toteuttaa hyödyntäen useita erilaisia topologioita. [7]

Topologia on periaatteeltaan kartta verkon rakenteesta. Topologiat voidaan jakaa kolmeen peruskategoriaan, joita ovat fyysiset topologiat (physical topologies), signaali topologiat (signal topologies) sekä loogiset topologiat (logical topologies). Fyysinen topologia kuvaa kaapeleiden ja laitteiden asettelun sekä sijainnin verkossa. Signaali topologian avulla voidaan esittää signaalien käyttämät todelliset reitit ja looginen topologia kuvaa tiedon käyttämät näennäiset yhteydet verkon solmupisteiden välillä. Verkolla voi olla eräänlainen fyysinen topologia ja täysin erilainen looginen topologia. Loogista topologiaa voidaan muokata dynaamisesti erilaisilla laitteilla, kuten kytkimillä ja reitittimillä. [1]

Väylätopologia

Väylätopologia on rakenteeltaan lineaarinen ja siihen liitetyt laitteet ovat kytketty ketjuttamalla sarjaan tai käyttämällä erilaisia haaroittimia. Signaalin heijastumisen estä-

miseksi väylärakenteen aloitus- ja lopetuspisteeseen kytketään päätevastus. Verkon resurssit jaetaan kaikkien kytkettyjen laitteiden kesken, mikä tekee väyläverkosta edullisen, mutta samalla rajoittaa verkon suorituskykyä sekä luotettavuutta. Tästä syystä yhteen väyläsegmenttiin kytkettävien laitteiden määrä on rajoitettu ja yleensä suhteellisen pieni. [7] Tietoliikennemielessä väylätopologian haittapuolena on verkossa kulkevan tiedon reititys jokaiselle väylään kytketylle laitteelle. Tämä lisää turvallisuusriskejä, kuten salakuuntelun mahdollisuuden sekä koko verkkosegmentin toiminnan menettäminen fyysisen kaapeloinnin vikaantuessa. [8]

Puutopologia

Puutopologia on rakenteeltaan hierarkkinen väylätopologian laajennos, jossa runkotopologia (trunk) tukee täydentäviä haaratopologioita (branches) [7, 9]. Runkotopologiasta haarautuvilla osilla voi olla lisäksi omia haaroja, joka mahdollistavat monimutkaisemmat rakenteet [9]. Puurakennetta käytetään esimerkiksi Foundation Fieldbus H1 -kenttäväylässä. Puutopologia H1-kenttäväylässä luodaan liittämällä väylätopologiaan haaroittimia, jotka mahdollistavat tähtikytkennän useille kenttälaitteille. [7]

Tähtitopologia

Tähtitopologia mahdollistaa useiden verkkolaitteiden kytkemisen yhteen keskitettyyn tähtipisteeseen. Perinteiset Ethernet-kytkimet tarjoavat mahdollisuuden kytkeä laitteita tähtipisteeseen. Tähtipisteeseen kytkettyihin päätepisteisiin voidaan liittää myös uusia kytkimiä ja luoda lisää tähtipisteitä. [7] Tähtitopologia mahdollistaa tietoliikenneyhteydet tiettyjen laiteryhmien välillä siten, että muut laitteet eivät ole tietoisia viestinnästä. Jos verkon haara vikaantuu, se ei vaikuta muun verkon toimintaan. Näistä syistä tähtitopologian turvallisuutta voidaan pitää korkeampana, kuin väylätopologian. [8] Tähtitopologian haittapuolena on koko verkon toimintakyvyn menettäminen keskitetyssä tähtipisteessä sijaitsevan laitteen vikaantuessa [10].

Rengastopologia

Rengastopologia on nimensä mukaisesti rakenteeltaan ympyränmuotoinen. Rengastopologiassa verkon laitteet ovat kytketty sarjaan, mutta viimeinen laite kytketään takaisin ensimmäiseen laitteeseen. Verkon päätepisteitä ei kytketä päätevastuksille, kuten aikaisemmin esitetyssä väylätopologiassa. Rengasrakennetta käytetään yleensä runkoverkkona järjestelmän kytkimille. [7] Rengastopologiassa jokaisella laitteella on vähintään kaksi yhteyspistettä verkkoinfrastruktuuriin. Katkokset renkaan osissa ei vaikuta muun verkon toimintaan. Haittapuolena voidaan pitää erityisesti rengastopologiaa varten suunniteltujen laitteiden tarvetta. [8]

Mesh-topologia

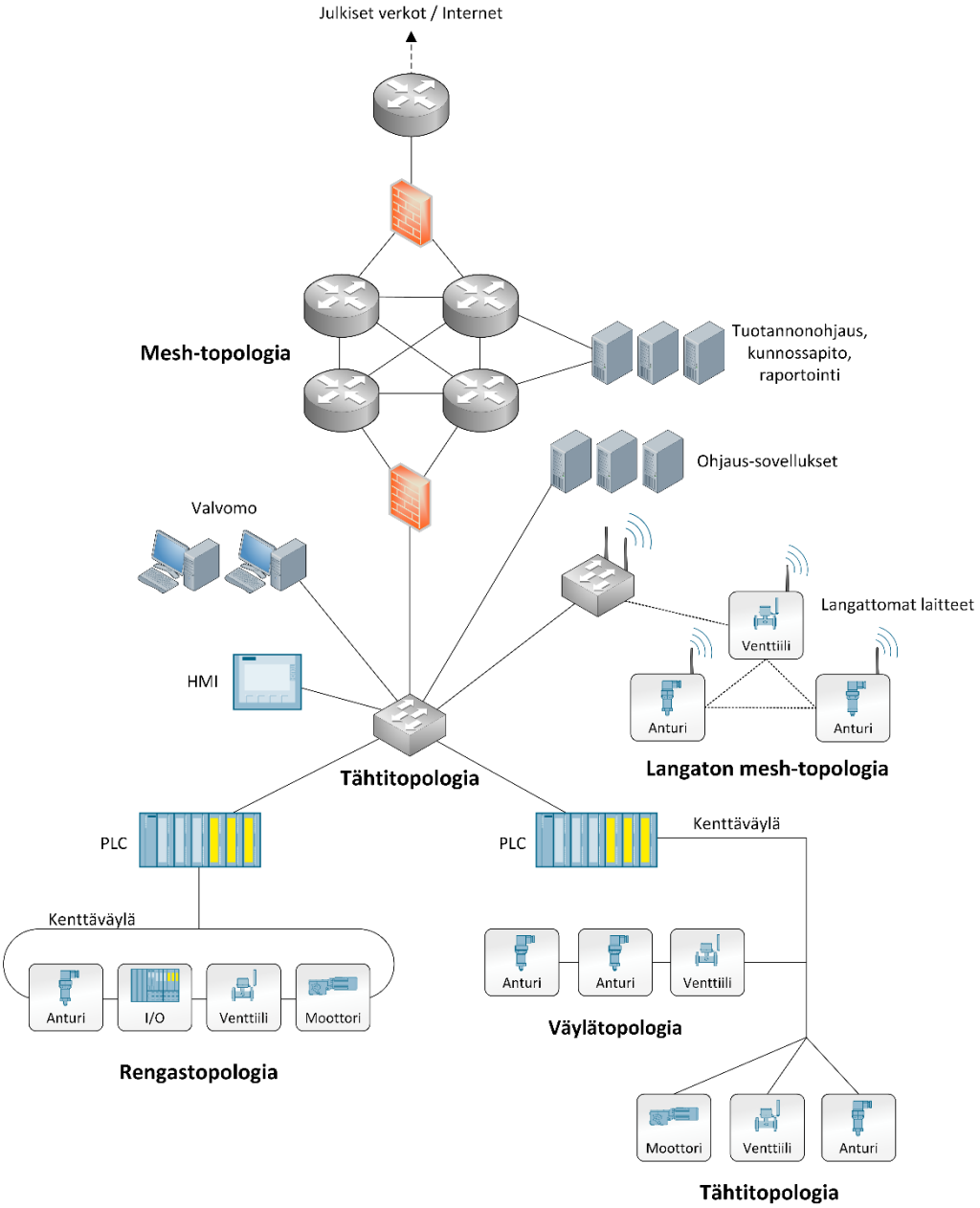
Mesh-topologia on yleisesti kriittisten laitteiden liitynnöissä käytetty rakenne, kun edellytetään korkeaa suorituskykyä sekä vikasietoisuutta. Rakennetta käytetään Ethernet-verkon runkolaitteiden, reitittimien sekä kriittisten palvelimien liitäntään. Verkon rakenne on toteutettavissa niin, ettei yksittäisen yhteyden tai laitteen vikaantuminen heikennä verkon suorituskykyä. Mesh-topologiaa käytetään yleisesti myös langattomien verkkojen rakenteena. [7]

Topologiat käytännössä

Järjestelmän fyysinen topologia voi rakentua monilla tavoin, mutta yksinkertaiset ja monimutkaisemmat rengastopologiat ovat yleisiä suurissa asennuksissa, kuten prosessiautomaatiossa. Linja- ja tähtitopologiat ovat myös yleisiä etenkin kappaletavaraautomaatiossa. Mesh-topologioita käytetään myös, mutta ne ovat monimutkaisempia hallita redundanttisuuden näkökulmasta. [11]

Nykyaikaisissa yritysverkoissa ei käytetä enää väylä- tai rengastopologioita, mutta ne ovat usein välttämättömiä teollisuuden viestintäjärjestelmissä. Rengastopologian avulla pystytään rakentamaan teollisuusverkoissa tarvittavat redundanttisuus ominaisuudet ja saavuttamaan näin korkeampi luotettavuus. Mesh-topologian toteuttaminen on nykyisin suhteellisen edullinen sekä erittäin tehokas menetelmä yritysten runkoverkkojen ja liiketoiminnan kannalta kriittisten palvelimien redundanttisuutta toteutettaessa. Teollisuusverkoissa on yleisempää rakentaa runkoverkko rengastopologialla ja käyttää tähtitopologiaa laitteiden yhdistämiseksi renkaassa oleviin kytkimiin. Teollisuusympäristössä, joka toteutetaan korkeamman luotettavuuden saavuttamiseksi langallisella teknologialla, mesh-topologian toteuttaminen rengas- tai väylätopologioiden tilalla kasvattaisi kustannuksia kohtuuttomasti. Mesh-teknologia on kuitenkin noussut standardiksi langattomissa teollisuusverkoissa. [7]

Siirryttäessä teollisuuden prosessitasolta lähemmäksi ylätason järjestelmiä, muuttuu teollisuusverkoissa käytetty rakenne vastaamaan perinteisiä yritysten datakeskuksia. Ylätason järjestelmien runkokytkimet ja reitittimet voidaan yhdistää toisiinsa käyttäen mesh-topologiaa. [7] Kuvassa 1 on esitetty esimerkki teollisuusverkosta, josta ilmenee tarve useille erilaisille verkkotopologioille.



Kuva 1. *Esimerkki teollisuusverkosta, joka koostuu useiden topologioiden ja laitteiden yhdistelmästä. Mukailtu lähteestä [7].*

2.3 Automaatioverkko

Automaatioverkolla tarkoitetaan järjestelmää, jossa toisiinsa liitettyjä laitteita käytetään ohjaamaan ja valvomaan fyysistä prosessia teollisuusympäristössä. Nämä verkot eroavat toiminnallisten erityisvaatimustensa takia merkittävästi perinteisistä yritysverkoista. Toiminnallisista eroista huolimatta automaatioverkkojen ja yritysverkkojen välillä on kasvavaa integraatiota. [12]

Digitaaliset ohjausjärjestelmät ovat verkottuneet jokaisella teollisuuden ohjausjärjestelmän tasolla. Ethernet-standardeja hyödyntämällä on saavutettu yritys- ja teollisuusverk-

kojen integraatioita. Tämä on johtanut verkkoympäristöihin, jotka mustuttavat tavanomaisia yritysverkkoja, mutta joilla on huomattavasti erilaiset vaatimukset. [12]

Automaatioverkko huolehtii kenttälaitteiden, digitaalisten ohjainten, erilaisten ohjelmistopakettien ja ulkoisten järjestelmien kommunikointiprotokollien implementoimisesta. Automaation lisääntyminen teollisuusympäristössä on jatkuvassa kasvussa ja tästä syystä teollisuuden verkkoja integroidaan entistä enemmän perinteisten tekniikoiden kanssa. Teollisuusverkkojen kehittäminen, käyttöönotto, käyttö sekä ylläpito vaativat erityisosaamista teollisten verkkojen perusperiaatteista, toiminnoista sekä vaatimuksista. [12]

2.3.1 Automaatio- ja yritysverkon vaatimukset

Automaatioverkkojen viimeaikaiset edistysaskeleet, kuten Ethernet-tekniikan lisääntynyt käyttöönotto, ovat hämärtäneet teollisten ja perinteisten verkkojen välistä rajaa. Näiden verkkojen välillä on kuitenkin keskeisiä eroja esimerkiksi vaatimuksissa. Yksi tärkeä ero automaatioverkon ja yritysverkon välillä on, että automaatioverkkoon liitetään fyysisiä laitteita, joita käytetään ohjaamaan ja valvomaan reaalimaailman toimintoja ja olosuhteita. Taulukossa 1 on esitetty teollisten- ja perinteisten verkkojen välisiä eroavaisuuksia. [12]

Taulukko 1. *Teollisten- ja perinteisten verkkojen välisten vaatimusten eroavaisuudet. Mukailtu lähteestä [12].*

	Teollisuusverkko	**Perinteinen verkko**
Ensisijainen toiminto	Fyysisten laitteiden ohjaus	Tiedonkäsittely ja -kuljetus
Sovellusalue	Valmistus, prosessiteollisuus sekä kriittinen infrastruktuuri	Yritys- ja kotitalousympäristö
Hierarkia	Syvä, toiminnallisesti erotetut hierarkiat, jotka sisältävät useita protokollia ja fyysisiä standardeja	Matala, integroidut hierarkiat, jotka sisältävät yhtenäisen protokollan sekä fyysisen standardin
Vikaantumisen vakavuus	Korkea	Matala
Vaadittava luotettavuus	Korkea	Kohtalainen
Vasteaika	250 µs – 10 ms	50+ ms
Deterministisyys	Korkea	Matala
Tiedon koostumus	Jaksottainen ja jaksoton liikenne, joka koostuu kooltaan pienistä paketeista	Jaksoton liikenne, joka koostuu kooltaan suurista paketeista
Ajallinen eheys	Vaaditaan	Ei vaadita
Operointiympäristö	Vaativat olosuhteet, jotka sisältävät usein pölyä, lämpöä, kosteutta sekä tärinää	Puhdas ympäristö, joka on tarkoitettu herkille laitteille

Implementointi

Teollisuusverkkoja käytetään useilla teollisuudenaloilla, kuten valmistavassa tuotannossa, sähköntuotannossa ja jakelussa, elintarvike- ja juomateollisuudessa, veden jakelussa, jäteveden käsittelyssä sekä kemiallisessa jalostamisessa. Teollisuusverkkoja tarvitaan lähes jokaisessa tilanteessa, jossa koneita halutaan valvoa ja ohjata. Jokaisella teollisuuden alalla on hieman toisistaan eroavat vaatimukset. [12]

Arkkitehtuuri

Teollisuuden verkoissa on yleisesti syvempi arkkitehtuuri, kuin yritysverkoissa [13]. Yritysverkot voivat koostua esimerkiksi haarakonttorien ja toimistojen lähiverkoista, jotka ovat yhdistetty toisiinsa runkoverkon tai laajaverkon avulla. Pienimmätkin teollisuusverkot ovat yleensä rakennettu kolmeen tai neljään tasoon. Esimerkiksi kenttälaitteiden ja ohjainten väliset yhteydet toteutetaan alimmalla tasolla, ohjainten väliset yhteydet toteutetaan toisella tasolla, järjestelmän valvontaan ja ohjaukseen käytettävät käyttöliittymät kolmannella tasolla sekä yhteydet tiedonkeruuta ja ulkoista kommunikointia varten neljännellä tasolla. Teollisuuden verkkoprotokollien ja teknologioiden kehittyminen on johtanut hierarkian sulautumiseen, etenkin ylimpien kerrosten yhdistymiseen. Verkkoarkkitehtuuria ei yleensä sulauteta niin paljon kuin olisi mahdollista, jotta säilytetään korrelaatio ohjauslaitteiden toiminnallisessa hierarkiassa. [12]

Vikaantumisen vakavuus

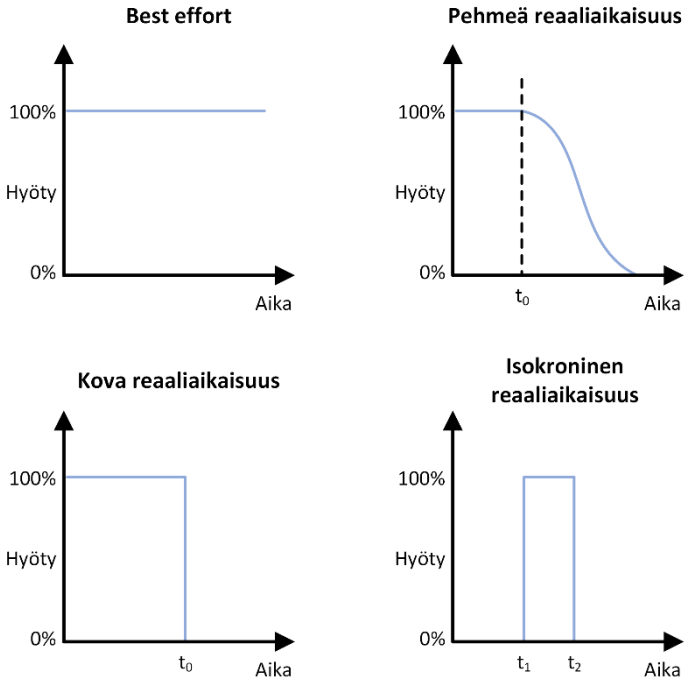
Automaatioverkot ovat yhteydessä fyysisiin laitteisiin ja järjestelmän vikaantumisella on vakavampia seurauksia kuin yrityksen järjestelmän vikaantumisella. Erilaiset verkon vikaantumisen vaikutukset ovat korostuneet, ne voivat sisältää laitevaurioita, tuotannollisia menetyksiä, ympäristövahinkoja, maineen menetyksen, henkilövahinkoja tai jopa kuolemaan johtavia tapaturmia. Vaikka vikaantuminen ei aina aiheudu ohjausjärjestelmän vikaantumisesta, lukuisat teollisuusonnettomuudet ovat esimerkkejä vakavan teollisen epäonnistumisen vaikutuksista. [12]

Reaaliaikavaatimukset

Prosessien ja laitteiden operointinopeus saattaa vaatia tiedon lähetyksen, käsittelyn sekä vastauksen tapahtuvan mahdollisimman nopeasti. Monissa liikkeenohjaussovelluksissa vasteaikavaatimukset ovat erittäin tiukkoja [13], jopa 250µs – 1ms ja vähemmän aikakriittiset prosessit saattavat vaatia 1ms – 10ms vasteaikaa. Tiedonsiirron viivästyminen voi vaikuttaa merkittävästi esimerkiksi säätöpiirien suorituskykyyn. [12] Yritysverkoissa ei yleensä ole vasteaikavaatimuksia [14], mutta jos niitä joissain tapauksissa on, ne ovat 10ms – 100ms luokkaa tai jopa useita sekunteja. Teollisuusverkkohierarkian ylimmät tasot pyrkivät pienentämään vasteaikavaatimuksia asteittain. [12]

Reaaliaikakäsite voidaan jakaa neljään ryhmään järjestelmältä vaaditun vasteajan perusteella. Nämä ryhmät ovat best effort, pehmeä, kova sekä isokroninen reaaliaika. Best effort tarkoittaa, että järjestelmällä ei ole varsinaista reaaliaikavaatimusta ja järjestelmän toteuttamat toiminnot ovat aina hyödyllisiä. [14] Pehmeässä reaaliaikavaatimuksessa järjestelmän tulee toteuttaa haluttu toiminto tiettyyn määräaikaan mennessä, mutta jos määräaika ylitetään ei siitä aiheudu järjestelmän virhettä ja palvelun toiminto on vielä joissain määrin hyödyllinen. Kovassa reaaliaikavaatimuksessa järjestelmän tulee toteuttaa haluttu toiminto tiettyyn määräaikaan mennessä. Järjestelmän ylittäessä reaaliaikavaatimuksen määräajan, järjestelmässä tapahtuu virhe tai palvelun toteuttama toiminto on hyödytön. [14, 15] Isokroonisessa reaaliaikavaatimuksessa haluttu toiminto tulee toteuttaa tietyn aikaikkunan sisällä. Aikaikkunan ulkopuolella suoritettu toiminto on hyödytön. [14]

Kuvassa 2 on esitetty reaaliaikaisen tehtävän tuottama hyötyä suhteessa vasteaikaan. Reaaliaikavaatimuksen ollessa kova, toiminnon tuottama hyöty laskee nollaan välittömästä määräajan ylittyessä. Reaaliaikavaatimuksen ollessa pehmeä, toiminnon tuottama hyöty alkaa pienentyä määräajan ylittyessä ja lähestyy nollaa ajan kuluessa. [14, 15] Isokroonisessa reaaliaikavaatimuksessa toiminnon hyöty on aikaikkunan ulkopuolella aina nolla [14]. Useissa reaaliaikajärjestelmissä on sekä kovia että pehmeitä reaaliaikavaatimuksia [15]. Perinteisissä informaatioteknologiaympäristöissä on tyypillisesti best effort tai pehmeä reaaliaikavaatimus. Teollisuuden automaatiojärjestelmien reaaliaikavaatimus voi olla myös kova tai isokroninen. Säätöpiireillä on tyypillisesti isokroonisia ja hälytyksillä kovia reaaliaikavaatimuksia. [14]



Kuva 2. *Erilaiset reaaliaikatyypit ja niillä saavutettava hyöty suhteessa vasteaikaan. Mukailtu lähteestä [14].*

Determinismi

Deterministisen verkon saavuttamiseksi on pystyttävä ennustamaan, milloin sanoma lähetetään ja vastaanotetaan. Tällä tarkoitetaan, että signaalin viiveen on oltava rajoitettu ja sen varianssin tulee olla matala. [12] Vasteajan varianssia kutsutaan jitteriksi (jitter). Matalaa jitteriä vaaditaan, koska viiveen vaihtelut vaikuttavat negatiivisesti esimerkiksi säätöpiirien toimintaan. [10, 12] Jitterin vaikutukset yritysverkossa eivät ole yhtä voimakkaita [12].

Tietopakettien koko

Teollisuusverkossa siirrettävät paketit ovat yleensä pieniä, varsinkin arkkitehtuurin alimmilla tasoilla [13], joissa siirretään yksittäisiä mittausarvoja tai digitaalisia tilatietoja. Tällaiset lähetykset ovat yleensä vain muutaman tavun kokoisia, kuten yksittäisen bitin tila tai 16-bittinen arvo. Pienien tietopakettien siirtäminen vaatii tarkoitukseen soveltuvia protokollia. Yritysverkossa lähetetään säännöllisesti paketteja, joiden koko on pienimmilläänkin satoja tavuja. [12]

Jaksollinen ja jaksoton liikenne

Teollisuusverkossa tarvitaan jaksollisesti näytteistettyä tietoa sekä jaksottomia tapahtumapohjaisia sanomia, kuten hälytyksen tai tilan muutos [10, 12]. Näytteenottojakso, jota käytetään tiedon keräämiseen ja lähettämiseen, voi vaihdella laitekohtaisesta tarpeesta riippuen. Jaksottomat sanomat voi esiintyä ajasta riippumatta, milloin tahansa. Teollisuusverkon tiedonsiirron oikea-aikaisuuden takaamiseksi kellot ja väylän kilpavarausprotokollat toteutetaan alhaisilla arkkitehtuuritasoilla. Yritysverkossa tiedonsiirto toteutetaan paras mahdollinen (best effort) -periaatteella, joka saattaa sisältää satunnaisen viiveen ennen tiedon lähettämistä. [12]

Eheys ja järjestys

Teollisuusverkossa on tarpeellista määritellä tapahtumien ajankohta ja järjestys. Nämä pystytään määrittelemään käyttämällä aikaleimausta sekä synkronisoituja kelloja. Järjestyksen ja ajallisen eheyden takaaminen ei yleensä ole osana yleisimpiä verkkoprotokollia, kuten TCP/IP:tä. [12] TCP/IP-viitemallia ja sen protokollapinoa käsitellään tarkemmin alaluvussa 3.2.

Ympäristö

Teollisuusverkkoja toteutetaan erilaisiin fyysisiin ympäristöihin, jotka ovat monesti haastavia. Ympäristön haasteet koostuvat kosteudesta, pölystä, lämpötilan vaihteluista sekä tärinästä. Laitteiden tulee kestää ympäristön asettamat vaatimukset virheettömän toiminnan takaamiseksi. [9, 12] Yritysverkoissa laitteet sijaitsevat kokonaisuudessaan puhtaissa ja lämpötilaltaan tasaisissa ympäristöissä [12].

3 TIEDONSIIRTOPROTOKOLLAT

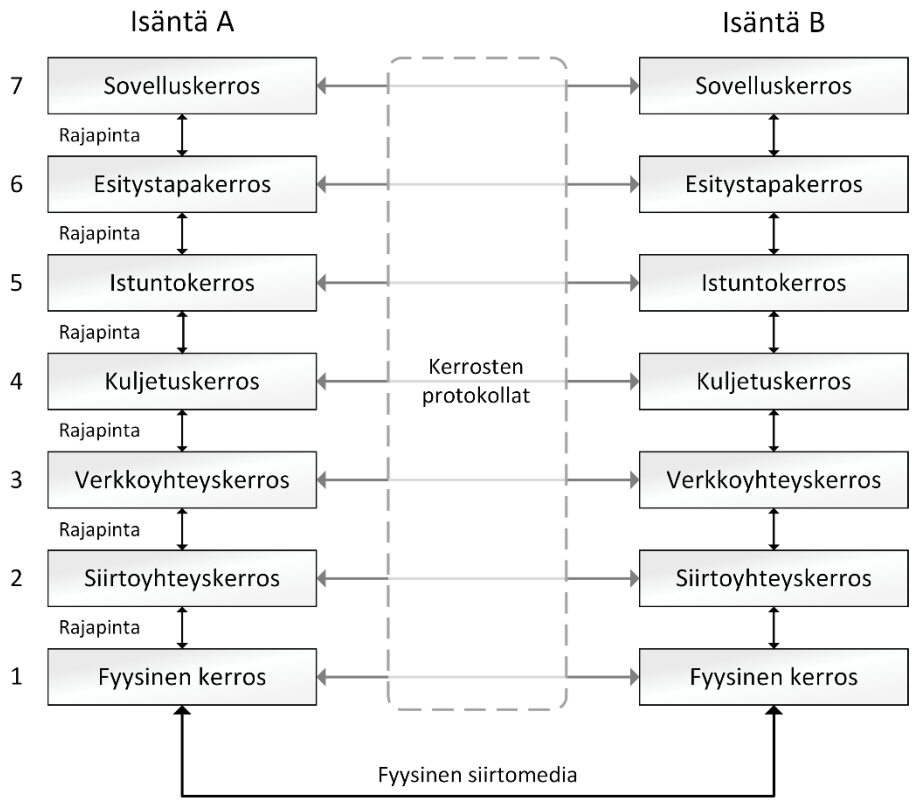
Tässä luvussa esitellään ensimmäiseksi tietoliikenneverkon kerrokset standardiviitemallien avulla. Viitemallit tarjoavat työkalut tietoliikenneverkkojen toiminnan kuvaamiseksi sekä ymmärtämiseksi. Seuraavaksi käsitellään automaation Ethernet-pohjaisia tiedonsiirtoprotokollia ja tarkastellaan niiden sijoittuminen kerrosmalleihin sekä TCP/IP-protokollapinoon.

Tiedonsiirtoprotokolla on joukko sääntöjä sekä määrittelyitä, joka mahdollistaa tehokkaan tiedonvaihdon kahden tai useamman kommunikointiyksikön välillä [4, 16]. Protokolla määrittelee sanoman rakenteen ja kommunikoinnin vaiheet sanomanvaihdossa sekä tapahtumien lähettämistä ja vastaanottamista koskevat toimet [4]. Tiedonsiirtoprotokollat määrittelevät yksityiskohdat, mukaan lukien toimenpiteet, jotka suoritetaan virheiden tai odottamattomien tilanteiden ilmetessä [17].

Abstraktio, jota käytetään keräämään protokollat yhtenäiseksi kokonaisuudeksi, tunnetaan nimellä kerrosmalli. Kerrosmalli kuvaa kuinka kommunikointiin liittyvät näkökannat voidaan jakaa pienempiin kokonaisuuksiin, jotka toimivat keskinäisessä yhteistyössä. Näistä pienemmistä kokonaisuuksista käytetään nimitystä kerros. Protokollien jako kerroksiin auttaa protokollien suunnittelussa sekä niiden käyttöönottamisessa. [17]

3.1 OSI-viitemalli

Kansainvälinen standardisoimisjärjestö ISO (International Organization for Standardization) on kehittänyt seitsemän kerroksisen OSI-viitemallin avointen järjestelmien välisten yhteyksien määrittelemistä varten [16, 18, 19]. OSI-viitemalli ei määrittele yksittäisiä palveluita tai protokollia, vaan tarjoaa viitekehyksen erilaisten standardien ja protokollien määrittelemisen perustaksi [10, 19]. Monimutkaisia verkkokokonaisuuksia voidaan yksinkertaistaa jakamalla kokonaisuudet kerroksiksi ja määrittelemällä kerrokset käytettävien protokollien avulla [18]. OSI-viitemallin seitsemänkerroksinen rakenne on esitetty kuvassa 3.



Kuva 3. *OSI-viitemallin seitsemän kerroksinen rakenne. Mukailtu lähteestä [18].*

OSI-viitemallin kolme alinta kerrosta ovat fyysinen kerros (physical layer), siirtoyhteyskerros (data link layer) sekä verkkoyhteyskerros (network layer) [4, 18]. Näiden kerrosten muodostamaa ryhmää kutsutaan yleisesti alakerroksiksi [20]. Alakerrokset ovat riippuvaisia fyysisestä verkosta ja tukevat järjestelmien välisiä yhteyksiä sekä niiden välistä tiedonsiirtoa [18]. Viitemallin kolme ylintä kerrosta ovat istuntokerros (session layer), esitystapakerros (presentation layer) sekä sovelluskerros (application layer) [4, 18]. Kolmen ylimmän kerroksen ryhmää kutsutaan isäntäkerroksiksi (host layers) [20]. Isäntäkerrokset ovat sovelluspainotteisia ja mahdollistavat loppukäyttäjien sovellusprosessien keskinäisen vuorovaikutuksen. Ala- ja yläkerroksien välissä oleva kuljetuskerros (transport layer), erottaa sovelluspainotteiset kerrokset kommunikointikerroksista [18]. Kuljetuskerros toimii alemman ja ylemmän ryhmän välisenä rajapintana [19].

Jokainen kerros suorittaa tarkasti määritellyn toiminnon. Tietovirrat kerrosten välillä kulkevat kerrosten rajapintojen lävitse. Kerrosten välinen sanomanvaihto tapahtuu käyttäen alemman kerroksen palveluita. Jokainen kerros kommunikoi suoraan etäjärjestelmän vastaavan kerroksen kanssa ja tarjoaa palveluita ylemmälle kerrokselle. Yksittäisen kerroksen toteutus on riippumaton muiden kerrosten toteutuksesta. [18] Seuraavaksi tarkastellaan lyhyesti jokaista OSI-viitemallin seitsemää kerrosta.

Fyysinen kerros

OSI-viitemallin alin kerros kuvaa toiminnallisen määrittelyn sekä kaapelointiin ja signaalin siirtoon liittyvät sähköiset ja mekaaniset ominaisuudet [2, 16, 18]. Sähköisiä ominaisuuksia voi olla esimerkiksi käytettävä jännitetaso, kaapelin resistanssi, signaalin pituus sekä bitin tilan ilmaiseva jännitetaso. Mekaanisiin ominaisuuksiin kuuluvat esimerkiksi liittimen koko ja muoto. [16, 18] Sarjamuotoiseen tiedonsiirtoon voidaan käyttää sähköistä, optista tai langatonta siirtomediaa [18]. Näitä ominaisuuksia tarvitaan yhteyden muodostamiseen, ylläpitämiseen sekä päättämiseen [16, 18]. Fyysinen kerros ei tarjoa virheenkorjauspalveluita, se voi kuitenkin tarjota tiedonsiirron ja virheiden määrän seurantaan liittyviä tietoja. Verkon fyysiset ongelmat, kuten vaurioitunut kaapelointi, vaikuttavat fyysisen kerroksen toimintaan. [16]

Siirtoyhteyskerros

Siirtoyhteyskerros on toinen kerros OSI-viitemallissa. Kerroksen tehtävänä on kontrolloida viestintää verkkoyhteyskerroksen ja fyysisen kerroksen välillä sekä tarjota luotettavuutta tiedonsiirtoon käytettäessä epäluotettavia fyysisiä medioita. [16] Kerroksen tarkoitus on siirtää sanomia, joita kutsutaan kehyksiksi (frames), fyysisen kerroksen lävitse. Se myös muodostaa yhteyden kahden suoraan toisiinsa kytketyn verkkolaitteen välille. Yksittäisessä verkkolaitteessa siirtoyhteyskerros toimii siirtomedian ja ohjelmiston yhdistävänä rajapintana. [4, 16]

Siirtoyhteyskerroksen päätehtävä on tunnistaa ja korjata tiedonsiirtovirheet, fyysisen kerroksen käsitellessä vain raakatiedonsiirtoa. Siirtoyhteyskerros ratkaisee ongelmia, jotka aiheutuvat kehyksien katoamisesta, vahingoittumisesta tai monistumisesta. [18] Siirtoyhteyskerros tarjoaa erilaisia palveluita, esimerkiksi mekanismeja verkkoliikenteen säätelemiseksi. Verkkoliikenteen nopeutta voidaan joutua säätelemään, jos vastaanottajan prosessointinopeus on hitaampi kuin lähettäjän. Prosessointinopeuden ero voi aiheuttaa kehysten katoamisen. [16, 18]

Verkkoyhteyskerros

Kolmas OSI-viitemallin kerros on verkkoyhteyskerros. Verkkoyhteyskerroksen tehtävä on huolehtia tietopakettien reitittämisestä verkkojen välillä lähettäjältä vastaanottajalle. [16, 18, 21] Verkkoyhteyskerros muodostaa loogisen yhteyden verkon laitteiden välille [16]. Kerros voi tarjota myös palveluita, kuten lähetysten priorisoinnin, vuonvalvonnan ja palvelun laadun varmistamisen. [16, 20, 21]

Tietopakettien reitityksessä voidaan käyttää staattista tai dynaamista reititystä. Reititys voidaan muodostaa lähetyksen alussa, jolloin kaikki tietopaketit kulkevat samaa reittiä lähettäjältä vastaanottajalle. Toinen vaihtoehto on muodostaa reititys jokaiselle tietopaketille erikseen. Reitityksen valintaan voi vaikuttaa verkon kuormitustilanne, jolloin voidaan välttää kuormittamasta lisää ruuhkautunutta verkon osaa. [18]

Verkkoyhteyskerrokseen voidaan sisällyttää mekanismeja verkkoliikenteen analysoimista varten. Mekanismien avulla lasketaan lähetettyjen ja vastaanotettujen pakettien määrä sekä kerätään tietoa lähettäjistä ja vastaanottajista. Kerättyjä tietoja voidaan käyttää verkonhallinnassa ja sen avulla voidaan koostaa esimerkiksi laskutuksessa tarvittavat tiedot. [18]

Kuljetuskerros

Kuljetuskerroksen tehtävä on huolehtia suoran järjestelmien välisen yhteyden muodostamisesta, eli tiedon läpinäkyvästä kuljettamisesta lähettäjän ja vastaanottajan välisessä verkossa [16, 21]. Kuljetuskerros toimii alempien, tiedon kuljettamisesta vastaavien sekä ylempien, sovellusprosessien välisestä kommunikoinnista vastaavien, kerrosten välisenä rajapintana [18, 21]. Kuljetuskerroksen tehtävistä vastaavat kuljetusprotokollat (transport protocols) [20].

Kuljetuskerroksen protokollien tehtävä on sovellusten lähettämän tietovirran pilkkominen sopivan kokoisiksi paketeiksi. Protokollien tehtävä on myös huolehtia yhteyden muodostamisesta ja purkamisesta asiakas- ja palvelinohjelmistojen välillä sekä tiedon eheyden varmistaminen sopivan kuittausmenettelyn avulla. [16, 18, 20] Tiedon pilkkominen, pakettikoon määritys sekä kuittausmenettely muodostavat kokonaisuuden, jota kutsutaan vuonohjaukseksi. Protokollia, jotka huolehtivat yhteyden muodostusrutiineista, pakettien koon määrittelystä sekä kuittausmenettelyistä, kutsutaan yhteydellisiksi protokolliksi (connection oriented protocols). Kaikki kuljetuskerroksen protokollat eivät ole yhteydellisiä protokollia. Yhteydettömät protokollat (connectionless protocols), huolehtivat tietovirran pilkkomisesta, mutta eivät vastaa muusta vuonohjauksesta. [20]

Istuntokerros

Istuntokerros huolehtii liikennöivien järjestelmien sovellusten välisien yhteyksien muodostamisesta, ylläpitämisestä sekä purkamisesta [16, 18, 21]. Istuntokerrokselle on määritetty tavanomaisista protokollatoiminnoista poikkeavia toimintoja [21], kuten käyttöoikeuksien tarkastaminen sekä muita järjestelmän suojaukseen liittyviä toimintoja. Kerroksen ohjelmistojen tehtävänä on tarjota kirjautumisrutiinit sekä salausmenetelmät. Nykyaikaisissa järjestelmissä useimmista tämän kerroksen tehtävistä huolehtii käyttöjärjestelmä. Salausohjelmistot sekä tietokantojen hallintajärjestelmät toimivat myös osittain tämän kerroksen ohjelmistoina. [20] Istuntokerros mahdollistaa kirjautumisen toisen laitteen resursseihin, esimerkiksi etäkäyttöä tai tiedostonjakoa varten. Kerros tarjoaa myös vuorovaikutuksenhallinnan sekä varmistaa tiedonsiirron tapahtuvan virheettömästi. [16, 18]

Esitystapakerros

Esitystapakerros määrittelee järjestelmien välisen sanomaliikenteen syntaksin. Kerros kuvaa tiedon esitysmuodon sovelluskerrokselta verkossa siirrettävään muotoon ja toi-

sinpäin. [16, 18, 20] Kerros toimii tiedon esitystavan muuntajana [18] ja sen määrittelyyn kuuluvat erilaiset koodausjärjestelmät [20]. Esitystapakerros voi tarjota palveluita myös tiedon salaukseen sekä pakkaamiseen [16, 18]. Nykyaikaisissa verkkojärjestelmissä tämän kerroksen tehtävistä huolehtii käyttöjärjestelmä [20].

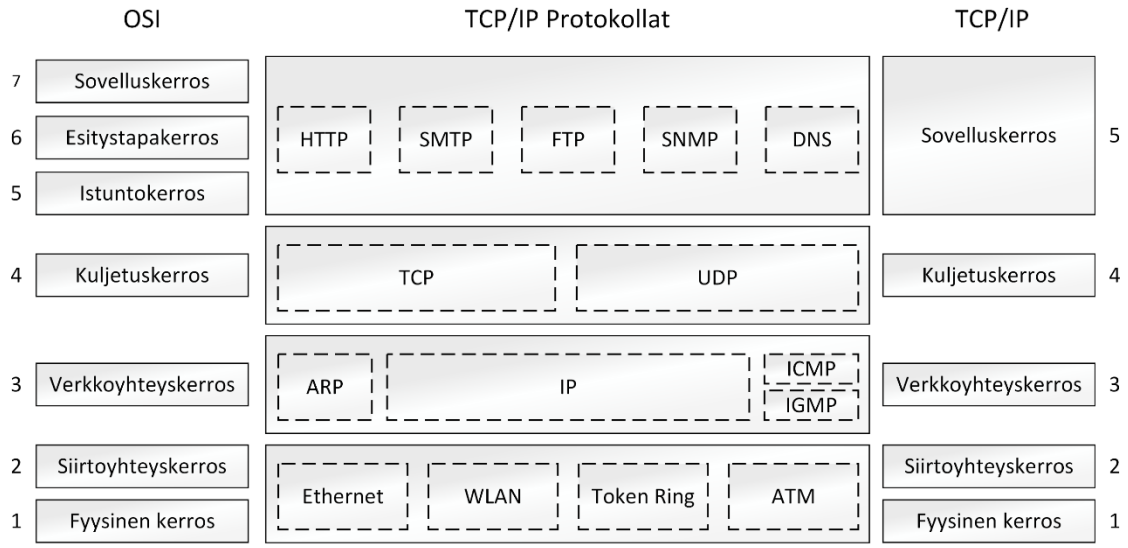
Sovelluskerros

OSI-viitemallin ylintä kerrosta kutsutaan sovelluskerrokseksi. Sovelluskerros kuvaa yhteydet käyttäjäsovellusten ja OSI-mallin välillä sekä tarjoaa rajapinnan tietoliikennesovellusten ja verkon palveluiden välille. [16, 21] Sovelluskerros ei kuitenkaan suorita varsinaisia ohjelmistoja verkossa, vaan se tarjoaa palveluita, kuten tiedostonsiirto, tiedostonhallinta ja sähköpostin tietojen käsittely [16, 18, 21]. Sovelluskerros tarjoaa lisäksi palveluita kommunikaatio kumppaneiden identifiointiin, kumppanin käytettävyyden tarkastamiseen, kommunikointi valtuuksien tarkastamiseen, tietosuojapalveluiden tarjoamiseen, kumppanin todentamiseen, virheiden korjausmenettelyiden sopimiseen sekä tiedon syntaksin rajoituksien tunnistamiseen [18].

3.2 TCP/IP-viitemalli

TCP/IP-viitemalli määrittelee joukon sääntöjä, jotka mahdollistavat laitteiden välisen kommunikoinnin verkon välityksellä. [1] Viitemalli on rakenteeltaan samanlainen kuin edellä esitelty OSI-viitemalli, mutta se sisältää vain viisi abstraktiokerrosta [1, 9, 16, 18]. TCP/IP-viitemallin kerrokset alhaalta ylöspäin lueteltuna ovat fyysinen kerros, siirtoyhteyskerros, verkkoyhteyskerros, kuljetuskerros sekä sovelluskerros [1, 4, 18, 21]. TCP/IP-viitemalli ei kuvaa erikseen OSI-viitemallin mukaisia esitystapa- ja istuntokerroksia, mutta näiden kerrosten tuottamat palvelut voidaan sisällyttää sovellusprosesseihin [18].

OSI-viitemallin kolmea pääkonseptia on sovellettu TCP/IP-viitemallissa. Jokainen kerros käyttää alemman kerroksen palveluita ja tuottaa tarkasti määritellyt palvelut ylemmälle kerrokselle. Kerroksen palvelut ovat käytettävissä rajapinnan avulla, joka määrittelee tarvittavat parametrit sekä tulokset, joita odotetaan palautettavan. Kerrokselle määritetyt protokollat kommunikoivat suoraan etäjärjestelmän vastaavan kerroksen kanssa, riippumatta taustalla olevan verkon rakenteesta. [18] TCP/IP-viitemallin protokollien hierarkkisuus on tarkoituksella suunniteltu kevyemmiksi kuin OSI-viitemallissa [1]. Kuvassa 4 on esitetty TCP/IP-viitemallin kerrokset, kerroksilla yleisesti käytetyt protokollat sekä TCP/IP-viitemallin ja OSI-viitemallin välinen riippuvuus.



Kuva 4. *OSI-viitemallin ja TCP/IP-viitemallin välinen riippuvuus sekä TCP/IP-protokollapino. Mukailtu lähteistä [18, 21]*

Fyysinen kerros

Fyysinen kerros määrittelee OSI-viitemallissa esitetyllä tavalla yksityiskohtaiset tiedot taustalla olevasta siirtomediasta sekä siihen liittyvästä laitteistosta. Kaikki sähköisiä ominaisuuksia, radiotaajuuksia sekä signaalia koskevat määrittelyt kuuluvat fyysisen kerroksen määrittelyyn. [17] TCP/IP-viitemalli ei määrittele tiettyä teknologiaa fyysisen kerroksen toteutukselle, vaan tukee useita teknologioita [1, 18].

Ethernetilla on useita erilaisia fyysisen kerroksen protokollia. Kuten aikaisemmin mainittiin, näitä protokollia ovat esimerkiksi kierretty parikaapeli, koaksiaalikaapeli ja valokuitu. Käytettävästä protokollasta riippuen, tiedonsiirto laitteelta toiselle fyysisen median välityksellä, tapahtuu eri tavalla. [4]

Siirtoyhteyskerros

Siirtoyhteyskerroksen protokollat määrittelevät kommunikoinnin yksityiskohdat ylempien kerrosten ohjelmistopohjaisten protokollien sekä fyysisen kerroksen laitteistopohjaisen toteutuksen välillä. [17] Palvelut voidaan toteuttaa lähes millä tahansa verkkoteknologialla, jonka avulla voidaan siirtää IP (Internet Protocol) -paketteja [1, 18]. Käytettäviä verkkoteknologioita voivat olla esimerkiksi Ethernet, Token Ring, ATM (Asynchronous Transfer Mode) tai WLAN (Wireless Local Area Network) [9, 18].

Verkkoyhteyskerros

TCP/IP-viitemallin kolmatta kerrosta kutsutaan verkkoyhteyskerrokseksi [1, 4] tai Internet kerrokseksi [1, 9]. Verkkoyhteyskerroksen päätehtäviin kuuluvat osoitteenmuodostus, pakettien reititys sekä virheiden raportointi. Lisäksi kerros tarjoaa palvelun pakettien pilkkomiseksi ja uudelleen muodostamiseksi. [1, 18] Tärkeimmät protokollat

verkkoyhteyskerroksella ovat IP (Internet Protocol), ARP (Address Resolution Protocol), ICMP (Internet Control Message Protocol) sekä IGMP (Internet Group Management Protocol) [18].

IP-protokolla huolehtii pakettien reitityksestä, IP-osoitteista sekä pakettien pilkkomisesta ja uudelleen muodostamisesta [1, 18]. Se on pakettikytkentäinen protokolla, joka perustuu best effort yhteydettömään arkkitehtuuriin [18]. IP-protokolla voi kuljettaa useiden erilaisten ylempien kerrosten protokollien tietoja [1]. ARP-protokollan tehtävä on muuttaa verkkoyhteyskerroksen osoite siirtoyhteyskerroksen osoitteeksi. Tämä tarkoittaa esimerkiksi loogisen IP-osoitteen muuttamista fyysiseksi Ethernetin MAC-osoitteeksi [18, 20]. Jotkut protokollat kuten ICMP ja IGMP on koottu IP-protokollan päälle, mutta suorittavat verkkoyhteyskerroksen toimintoja [1]. ICMP-protokollaa käytetään lähettämään tiedonsiirtoa koskevaa diagnostiikkatietoa [18, 21]. IGMP-protokollaa käytetään ryhmälähetystietojen hallintaan [1, 18].

Kuljetuskerros

Neljäs kerros TCP/IP-viitemallissa on kuljetuskerros. Kuljetuskerros tarjoaa päästäpäähän tiedonsiirto ominaisuudet, jotka ovat riippumattomia taustalla olevasta verkosta. Virheidenhallinta, sanomien pilkkominen sekä vuonohjaus ovat kerroksen tuottamia palveluita. [1, 18] TCP/IP-viitemallissa on kaksi kuljetuskerroksen protokollaa [9]. TCP (Transmission Control Protocol) -protokolla tarjoaa luotettavan yhteydellisen kommunikointipalvelun ja takaa, että tieto vastaanotetaan niin kuin se on lähetetty. UDP (User Datagram Protocol) -protokolla on epäluotettava yhteydetön kuljetuskerroksen protokolla. Protokollaa käytetään nopean pakettien toimituksen ollessa lähetyksen virheettömyyttä tärkeämpää. [1, 9, 18]

Sovelluskerros

TCP/IP-viitemallin ylin kerros on sovelluskerros, jonka tehtävänä on tuottaa palveluita sovellusprosessien tarpeisiin. Sovelluskerros käyttää kuljetuskerroksen palveluita ja mahdollistaa verkkolaitteiden sovellusten välisen kommunikoinnin useiden erilaisten korkeantason protokollien avulla. [1, 9, 18] Sovelluskerroksen protokollat käsittelevät kuljetuskerroksen palveluita mustana laatikkona, joka tarjoaa vakaan verkkoyhteyden kommunikointia varten [1]. Sovelluskerroksella toimivia protokollia ovat esimerkiksi hypertekstin siirtoprotokolla HTTP (Hypertext Transfer Protocol), tiedostonsiirtoprotokolla FTP (File Transfer Protocol), sähköpostipalvelimen kommunikointiprotokolla SMTP (Simple Mail Transfer Protocol), Telnet yhteysprotokolla pääteyhteyksille, nimipalvelujärjestelmä DNS (Dynamic Name Service) sekä verkonhallinnassa laajasti käytetty hallintaprotokolla SNMP (Simple Network Management Protocol) [1, 9, 18]. Näiden perusprotokollien lisäksi on toteutettu lukuisia muita sovelluskerroksella toimivia protokollia [18]. TCP/IP-viitemalli ei määrittele OSI-viitemallin mukaisia kerroksia kul-

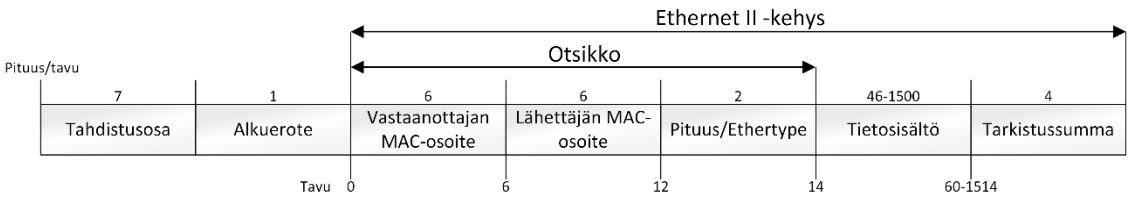
jetus- ja sovelluskerroksen välille, joten näiden kerrosten toteuttamat palvelut ovat tarvittaessa sisällytettävä sovelluskerroksen palveluihin [1, 9].

3.3 Verkkoarkkitehtuuri automaatiossa

Teollisuuden ohjausjärjestelmien arkkitehtuurit käyttävät tyypillisesti yhtä tai useampaa erityistarkoitukseen suunniteltua protokollaa. Arkkitehtuuri voi koostua patentoiduista valmistajakohtaisista protokollista sekä patentoimattomista avoimista protokollista kuten Modbus, CIP (Common Industrial Protocol) sekä Profibus. Useimmat näistä teollisuuden protokollista ovat alun perin suunniteltu sarjaliikenteellä toimiviksi, mutta nykyään ne ovat sovitettu toimimaan myös standardi Ethernetissä käyttäen verkkoyhteyskerroksella IP-protokollaa sekä UDP ja TCP -kuljetuskerroksen protokollia. Ethernet-pohjaiset versiot ovat yleisesti käytettyjä tämän päivän teollisuuden verkkoinfrastruktuureissa. [7]

Industrial Ethernet on laajempi kokonaisuus kuin perinteinen Ethernet-teknologia. Seitsemänkerroksisen OSI-viitemallin osalta tavallinen Ethernet-teknologia viittaa fyysiseen- ja siirtoyhteyskerrokseen, kun useimmat Industrial Ethernet ratkaisut käsittävät myös verkkoyhteyskerroksen sekä kuljetuskerroksen. Industrial Ethernet ratkaisut voidaan toteuttaa TCP/IP-protokollaperheen avulla tai sen rinnalla. [9] Kenttäväylän tietorakenne on sovitettu OSI-viitemallin ylimpiin kerroksiin tai TCP/IP-viitemallin mukaisesti sovelluskerrokseen [7, 9].

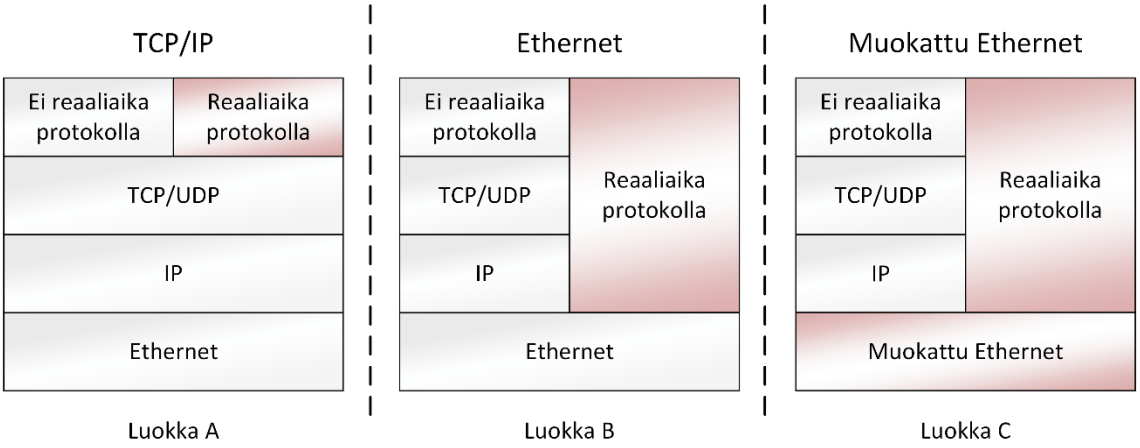
Industrial Ethernet-teknologia on pääsääntöisesti yhteensopiva perinteisten verkkolaitteiden ja verkkoinfrastruktuurien kanssa. IEEE-standardeihin perustuvien teknologioiden avulla automaatio- ja ohjaustoimintojen integroiminen Ethernet-ympäristöön helpottuu. Tällaisessa verkossa kytkimet toimivat siirtoyhteyskerroksella, käyttäen MAC-osoitteita. Industrial Ethernetissa kenttäväyläkohtaiset tiedot, joita käytetään I/O-laitteiden sekä muiden kenttälaitteiden ohjaukseen, on sulautettu Ethernet-kehyksiin (frames). [9] Kuvassa 5 on esitetty IEEE 802.3 standardin mukainen Ethernet-kehys. Ethertype-kentän avulla ilmaistaan, mikä protokolla on koteloitu Ethernet-kehyksen tietosisältökenttään. Esimerkiksi IP-protokollan Ethertype-merkintä on 0x800 ja Profinet-protokollan Ethertype-merkintä on 0x8892. [22]



Kuva 5. *IEEE 802.3 standardin mukainen Ethernet II -kehys. Mukailtu lähteestä [22].*

3.3.1 Käytännön toteutukset

Standardi Ethernet ei kykene saavuttamaan kaikkia reaaliaikavaatimuksia varsinkaan automaatioverkon kenttälaitetasolla tai liikkeenohjaussovelluksissa. Markkinoilla on olemassa erilaisia teknologiaratkaisuja reaaliaikaisuuden toteuttamiseksi, joko standardi Ethernetin tai muokatun Ethernetin avulla. Kaikille Ethernet-verkoille yhteistä on yleiskäyttöinen kaapelointi. [11] Kuvassa 6 esitetään kommunikointiprotokollien yksinkertaistettu rakenne ja jäsentely eri kerroksille.



Kuva 6. *Arkkitehtuurit kommunikointiprotokollien reaaliaikaisuuden toteuttamiseksi automaatiossa. Mukailtu lähteestä [11]*

Sovellukset, joilla ei ole reaaliaikavaatimuksia, käyttävät Ethernet-protokollia ISO 8802-3 standardissa määritetyllä tavalla sekä TCP/IP-pinon mukaisia verkko- ja kuljetuskerroksen protokollia [11]. Reaaliaikavaatimuksia sisältävien sovelluksien rakentamiseksi on olemassa kolme erilaista lähestymistapaa. Lähestymistavat ovat jaettu kuvan 6 mukaisesti kolmeen eri luokkaan A, B ja C. [23]

Luokan A lähestymistapa on säilyttää TCP/IP-viitemallin mukainen protokollapino ja toteuttaa reaaliaikaominaisuudet puhtaasti sovelluskerroksella [11, 23]. Reaaliaikasuorituskykyä rajoittaa verkkoinfrastruktuuri kuten verkkolaitteiden aiheuttama viiveiden vaihtelu. Tämä lähestymistapa täyttää lähinnä best effort -reaaliaikavaatimuksen. Joissain toteutuksissa muutoksia tehdään myös TCP/IP-pinossa, paremman suorituskyvyn saavuttamiseksi. [23] Luokan B lähestymistapa on rakentaa reaaliaikaominaisuudet standardi Ethernetin päälle TCP/IP-pinon rinnalle. Tässä luokassa voidaan usein käyttää standardi Ethernet-verkkoinfrastruktuuria sekä verkkolaitteita, mutta TCP/IP-pinoa ei käytetä reaaliaikaisessa kommunikoinnissa. Luokan C lähestymistapa on muokata Ethernet mekanismeja ja infrastruktuuria, jolloin voidaan saavuttaa kovan tai isokroonisen reaaliaikavaatimuksen mukainen suorituskyky. Fyysisen- ja siirtoyhteyskerroksen muutokset vaativat useimmissa tapauksissa muutoksia verkkoinfrastruktuuriin sekä tarkoitukseen soveltuvia verkkolaitteita. [11, 23]

3.3.2 Automaatioverkon profiilit

Kaikki teollisuusprotokollat ovat määritelty kansainvälisen standardointiorganisaation IEC (International Electrotechnical Comission) standardissa IEC 61158. Dokumentti on rakenteeltaan OSI-viitemallin mukainen. Dokumentti koostuu liitteen A taulukon 1 mukaisesti seitsemästä osasta. Kaikki verkot määritetään tyypeittäin standardin osissa 2-6 ja erilaisia tyyppejä on olemassa 24 kappaletta.

Standardissa IEC 61784 kommunikointiprofiilit ovat kerätty ryhmiksi ja jaettu liitteen A taulukon 2 mukaisesti. Profiilit luokitellaan kommunikointiprofiiliperheittäin, CPF (Communication Profile Family), niiden tunnistamiseksi. Profiiliperheet ovat listattu liitteen A taulukossa 3. Jokainen profiiliperhe voi vapaasti määritellä kommunikointiprofiilijoukot, CP (Communication Profile). Standardin määrittelemät kommunikointiprofiiliperheet, näihin kuuluvat kommunikointiprofiilit sekä profiileiden verkon fyysisen-, siirtoyhteys- sekä sovelluskerroksen määrittävät tyypit on esitetty liitteen A taulukossa 4. [11]

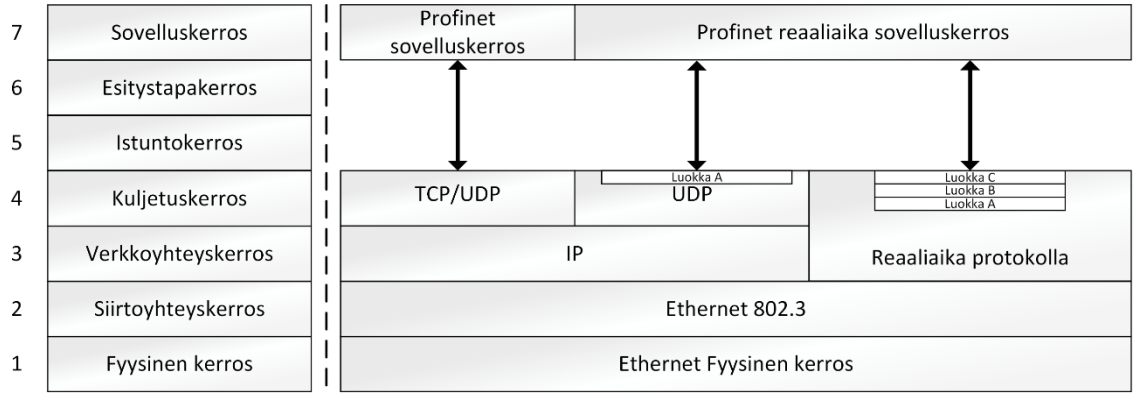
Profinet (Profiili 3/4-6)

Profinet on avoin Industrial Ethernet-standardi, jota ylläpitää PI (Profibus and Profinet International) -yhteisö. Profinet on useiden valmistajien yhteistyönä määrittelemä protokolla. [9, 11] Profinet mahdollistaa ohjainten ja kenttälaitteiden välisen kommunikoinnin Ethernet-teknologian avulla [22]. Profinet IO -kommunikointiteknologiat pohjautuvat laajasti hyväksyttyyn Profibus DP (Distributed Periphery) -protokollaan, joka on standardoitu IEC 61784-1 standardissa profiilina 3/1. Profinet IO -kommunikointiteknologiat ovat standardoitu kolmessa osassa edellä mainitussa standardissa, nämä teknologiat ovat Profinet IO Class A (Profiili 3/4), Profinet IO Class B (Profiili 3/5) sekä Profinet IO Class C (Profiili 3/6). [11]

Profinet IO -kommunikointiteknologiat hyödyntävät kytkentäistä full-duplex Ethernet-teknologiaa ja tukevat 100Mb/s tiedonsiirtonopeutta [24]. Profinet IO -kommunikointiprofiilit tukevat alaluvussa 2.2 esitellyistä verkkotopologioista väylä, tähti, rengas sekä puu -topologioita. Profinet IO -profiileiden fyysisenä mediana voidaan käyttää kuparikaapelia tai optista kuitua. Poikkeuksena Profinet IO Class A voidaan toteuttaa myös langattoman teknologian avulla, kuten Bluetooth tai WLAN -teknologiaa hyödyntäen. [24]

Profinet IO Class A -protokolla tarjoaa reaaliaikaominaisuuden, joka rakentuu UDP-kuljetusprotokollan tai reaaliaikaprotokollan päälle [22]. Profinet IO Class A -protokolla soveltuu jaksottaiseen (cyclic) sekä jaksottomaan (acyclic) tiedonsiirtoon [22, 24]. Protokolla ei aseta erityisvaatimuksia käytettäville verkkokytkimille [22]. Profinet IO Class B -protokolla tarjoaa reaaliaikaominaisuuden, joka rakentuu reaaliaikaprotokollan päälle. Protokolla soveltuu jaksottaiseen sekä jaksottomaan tiedonsiirtoon.

[22, 24] Protokolla asettaa erityisvaatimuksia käytettävälle verkkokytkimelle, mutta ei vaadi kommunikoinnin määrittelemistä laitteiden konfiguroinnin yhteydessä [22]. Profinet IO Class C -protokolla tarjoaa isokroonisen reaaliaikaominaisuuden, joka rakentuu reaaliaikaprotokollan päälle. Protokolla mahdollistaa jaksottaisen tiedonsiirron isokroonisissasovelluksissa. [22, 24] Protokolla asettaa erityisvaatimuksia käytettäville verkkokytkimille sekä vaatii kommunikoinnin määrittelemistä laitteiden konfiguroinnin yhteydessä [22]. Kuvassa 7 on esitetty Profinet IO -protokollat ja niiden asettuminen standardiviitemalliin.



Kuva 7. *Profinet IO -protokollat ja niiden asettuminen standardiviitemalliin. Mukailtu lähteestä [22].*

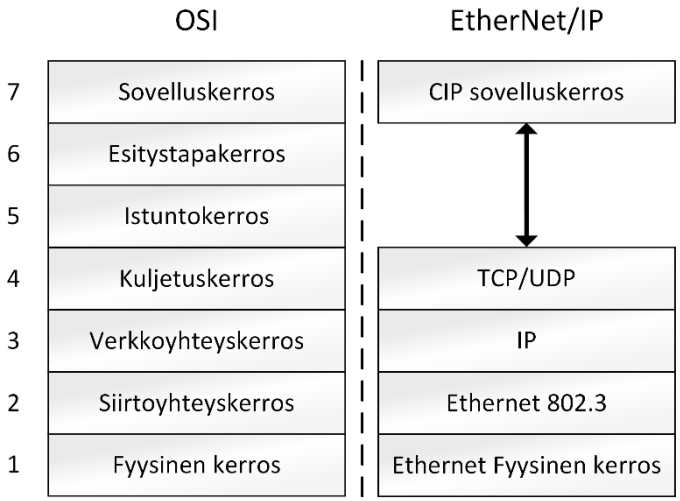
EtherNet/IP (Profiili 2/2)

EtherNet/IP-protokolla on kehitetty Ethernet-teknologiaan pohjautuvia automaatio- ja ohjausjärjestelmiä varten. Protokolla on Rockwellin määrittelemä ja sitä ylläpidetään ODVA (Open DeviceNet Vendors Association) organisaation toimesta. [9, 11] EtherNet/IP-kommunikointiteknologia on standardoitu IEC 61784-1 standardissa profiilina 2/2 [11]. EtherNet/IP perustuu CIP-protokollaan, joka on yhteinen sovelluskerroksen protokolla EtherNet/IP:lle, ControlNet:lle sekä DeviceNet:lle [9, 11].

CIP-protokolla on mediariippumaton, yhteyspohjainen sekä objektiorientoitunut protokolla. Se sisältää laajan valikoiman viestintäpalveluita automaatiosovellusten tarpeisiin, esimerkiksi ohjaukseen, turvallisuuteen, synkronisointiin sekä liikkeenohjaukseen. [25] CIP-protokolla määrittelee objektit, joiden avulla liitytään ohjattavan järjestelmän tietoihin, kuten laitteiden lähtöihin ja tuloihin, konfigurointi parametreihin sekä diagnostiikkatietoihin [11]. CIP-protokollaan on lisätty laajennoksia kovan reaaliaikavaatimuksen sovelluksia varten. [9]

EtherNet/IP on puhtaasti ohjelmistopohjainen ratkaisu [9]. EtherNet/IP mahdollistaa standardi Ethernet-teknologian ja TCP/IP-protokollapinon käyttämisen teollisuusautomaation sovelluksissa. Kuvassa 8 on esitetty EtherNet/IP-protokolla standardiviitemallin muodossa. IEEE-standardien noudattaminen mahdollistaa alaluvussa 2.2 esitettyjen verkkotopologioiden käyttämisen väyläverkon rakentamisessa. [26-28] Todellisuudessa

verkkotopologiaa rajoittaa EtherNet/IP-laitteiden tukemien loogisten yhteyksien määrä. EtherNet/IP erittelee TCP ja CIP -yhteydet. Useimmat laitteet tukevat 64 TCP-yhteyttä, mutta CIP-yhteyksien määrä vaihtelee. RockWellin mukaan yhteyksien enimmäismäärä vaihtelee 32-160 kyseessä olevasta laitteesta riippuen. [28]

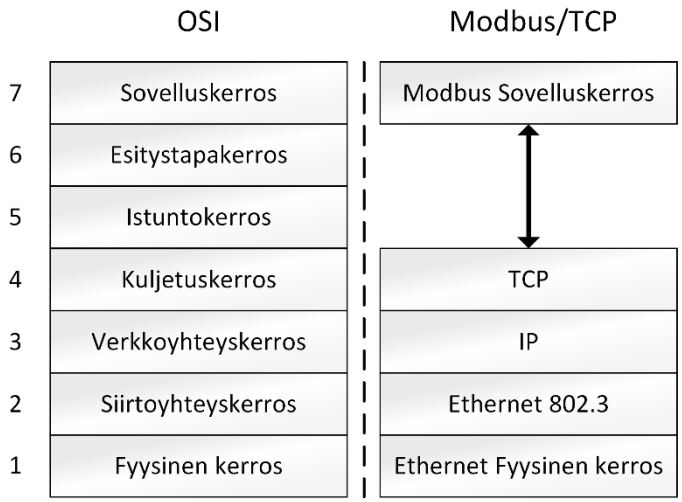


Kuva 8. *CIP-protokollaan perustuvan EtherNet/IP-protokollan sijoittuminen standardiviitemalliin. Mukailtu lähteestä [25].*

Suurin osa EtherNet/IP kommunikaatiosta muodostuu yleislähetys (broadcast) ja ryhmälähetys (multicast) -sanomista. Verkkokytkin ei voi välittää näitä sanomia vain yhteen porttiin. [28] Tämä johtaa helposti jonojen muodostumiseen kytkimissä, eli toisin sanoen aiheuttaa ennustamatonta viivettä [11]. Tämän vuoksi EtherNet/IP-verkoissa käytettävien kytkinten on tuettava Internet-ryhmien hallintaprotokollan snooping-teknologiaa (IGMP-snooping). Teknologia rajoittaa sanomatulvien määrää, määrittämällä dynaamisesti kytkimen porttien IP-monilähetysryhmät. Lisäksi on suositeltavaa, että kytkimet tukevat porttien peilausta, virtuaalisia lähiverkkoja sekä SNMP-protokollaa. [28]

Modbus/TCP (Profiili 15/1)

Modbus on nykyisin Schneider Electricin kehityksessä [11] ja Modbus/TCP on perinteisen Modbus-perheen laajennus Ethernet-ympäristöön [9]. Modbus/TCP on standardiviitemallien sovelluskerroksella toimiva protokolla [9, 29], joka on määritelty IEC 61784 standardissa kommunikointiprofiilina 15/1. Modbus/TCP on tänä päivänä yksi yleisimmin käytetyistä Ethernet ratkaisuista teollisuuden sovelluksissa. [11] Kuvassa 9 on esitetty standardiviitemallin sovelluskerroksella toimiva Modbus/TCP-protokolla.



Kuva 9. *Sovelluskerroksella toimivan Modbus/TCP-protokollan sijoittuminen standardiviitemalliin. Mukailtu lähteistä [7, 29].*

Modbus/TCP on hyvin yksinkertainen protokolla, joka perustuu kysely-vastaus -menetelmään (request-reply) ja tarjoaa mekanismit objektien lukemiseksi sekä muokkaamiseksi [11, 29]. Modbus/TCP ei aseta erityisiä vaatimuksia käytettäville verkkolaitteille ja topologioina voidaan käyttää kaikkia alaluvussa 2.2 esitettyjä verkkotopologioita [30]. Modbus/TCP ei takaa toimitusaikoja eikä tarjoa kovan reaaliaikavaatimuksen täyttävää ratkaisua. Modbus/TCP-kyselyt voidaan suorittaa toteutuksesta riippuen, joko yhdelle laitteelle kerrallaan, tai suorittaa kyselyitä useille laitteille rinnakkain. Rinnakkain suoritetut kyselyt parantavat kommunikoinnin suorituskykyä. Suorituskyky riippuu myös laitteiden TCP/IP-protokollapinon toteutuksesta. Parhaimmillaan voidaan päästä muutaman millisekunnin vasteaikoihin. [23]

Vaikka Modbus/TCP ei tarjoa kovan reaaliaikavaatimuksen täyttävää ratkaisua se on saanut laajennoksen, jonka avulla reaaliaikaominaisuuksia voidaan parantaa. Reaaliaikalaajennos on määritelty kommunikointiprofiilina 15/2. Modbus/TCP laajennos käyttää reaaliaikaista julkaisija-tilaaja -menetelmää, RTPS (Real-Time Publisher Subscriber), joka on suunniteltu toimimaan UDP-kuljetusprotokollan välityksellä. [11]

4 VERKONHALLINTA

Tässä luvussa esitellään verkonhallintaan liittyviä viitekehyksiä. Viitekehysten avulla etsitään verkon kunnonvalvonnan suuntaviivat sekä tarkastellaan verkonvalvontaa ja sen vaatimuksia. Lopuksi perehdytään verkonvalvontaan soveltuviin protokolliin sekä virtausteknologioihin.

Verkonhallinta on olennainen osa verkon luotettavaa toimintaa [31]. Se mielletään usein ylimääräiseksi osaksi verkon normaalin toiminnan rinnalle. Kun ongelmia ilmenee, mietitään, miksi ei ole olemassa helppoa keinoa selvittää, mitä tapahtui ja miksi. [32] Verkonhallinnan tarve korostuu yrityksien ollessa yhä riippuvaisempia verkosta ja sen avulla tuotetuista palveluista. Verkonhallinnan avulla pyritään varmistamaan verkon ja sen tarjoamien palveluiden oikeanlainen toiminta. Verkonhallinta auttaa pitämään verkon kustannukset hallinnassa. Verkonhallinnan avulla on mahdollista myös kasvattaa verkolla tuotettavaa arvoa. [31]

Laajojen tietokoneverkkojen hallinta on suuri tekninen haaste. Nykyiset tietoliikenneverkot voivat koostua sadoista tai tuhansista verkkolaitteista, kuten reitittimistä, kytkimistä, tukiasemista sekä palvelimista. Verkon ylläpitäjän pitää pystyä seuraamaan verkon tilaa ja hallitsemaan laitteita. Verkonhallintajärjestelmät on luotu verkon ylläpitäjän jokapäiväisen työn tueksi. Verkonhallintajärjestelmä on laitteiston ja ohjelmiston yhdistelmä, jonka avulla verkkoa voidaan hallita sekä valvoa. [33]

Verkkoresurssien valvonta, kuten palvelimien suorituskyvyn valvonta, on osa verkonhallintaa. Verkon- ja järjestelmänhallinnan raja on poistunut ajan saatossa ja niiden katsotaan olevan samaa kokonaisuutta. [34] Laitteiden hallinta ja valvonta jakautuu FCAPS (Faults, Configuration, Accounting, Performance and Security) -viitekehyksen mukaisiin toimintoihin. Toimintoihin kuuluvat verkon kokoonpanon havainnointi ja seuranta, laitteiden kunnon ja tilan valvonta, hälytyksien tarjoaminen verkon suorituskyvyn muutoksista sekä ongelmien tunnistus, identifiointi ja ratkaiseminen. [33] Verkonhallintajärjestelmä käyttää näiden toimintojen suorittamiseksi verkonhallinta- ja verkonvalvontaprotokollia, joita käsitellään alaluvussa 4.5.

4.1 Viitekehykset

Viitekehyksien tarkoitus on jakaa verkonhallinta sopiviksi kokonaisuuksiksi. Verkonhallinnan käsitteleminen pienempinä kokonaisuuksina on yksinkertaisempaa, kuin yhden ison kokonaisuuden käsitteleminen. Yritys voi määritellä tarpeidensa mukaisesti, minkälaisen painoarvon se antaa kullekin osa-alueelle. Viitekehykset eivät ota kantaa

varsinaiseen toteutukseen, vaan toteutuksesta tulee huolehtia yrityksen painottamien osa-alueiden mukaisesti. Viitekehyksien avulla määritetään toteutuksessa tarvittavat suuntaviivat. [31]

Alaluvussa 4.1.1 esitellään TMN (Telecommunication Management Network) -viitekehys, joka käsittelee verkonhallintaa laajassa merkityksessä yksittäisen verkkolaitteen hallinnasta aina yrityksen liiketoiminnan hallintaan asti. Alaluvussa 4.1.2 käsitellään FCAPS-viitekehystä, joka on TMN-viitekehykseen vertikaalisesti sidoksissa oleva toimintamalli. Edellä mainittujen viitekehyksien lisäksi on olemassa joukko muita vastaaviin tarkoituksiin suunniteltuja viitekehyksiä. TMN-viitekehyksen kanssa samanlaisia osa-alueita pyritään ratkaisemaan eTOM (enhanced Telecom Operations Map) -viitemallilla. Myös FCAPS-viitekehykselle on olemassa vaihtoehtoja, kuten OAM&P (Operations, Administration, Maintenance and Provisioning) sekä FAB (Fulfillment, Assurance, and Billing) -viitekehykset. [31]

TOM (Telecoms Operation Map) -viitekehyksen keskiössä sijaitsee elinkaarenhallinta. Viitekehys erottelee kolme elinkaaren vaihetta, jotka ovat toteuttaminen (fulfillment), varmistaminen (assurance) sekä laskutus (billing). Näistä vaiheista käytetään nimitystä FAB-viitekehys. Viitekehyksen jokaisella vaiheella on yksilöllinen joukko hallintavaatimuksia. Elinkaaren vaiheita käytetään TOM-viitekehyksen määrittelemillä kerroksilla. Nämä kerrokset ovat verkon- ja järjestelmänhallinta (network and systems management), palveluidenkehitys ja -käyttö (service development and operations) sekä asiakaspalvelu (custom care). [31]

TOM-viitekehyksestä on olemassa päivitetty versio, joka on nimeltään eTOM. eTOM-viitekehys laajentaa TOM-viitekehyksen soveltamisalaa sisällyttämällä tähän myös liiketoiminnan johtamisen osa-alueet. Osa-alueiden avulla pyritään huomioimaan erilaiset liiketoiminnan näkökohdat, kuten toimitusketjun hallinta, henkilöstöhallinta sekä taloudenhallinta. FAB-viitekehyksen osalta määrittely ei ole muuttunut. [31]

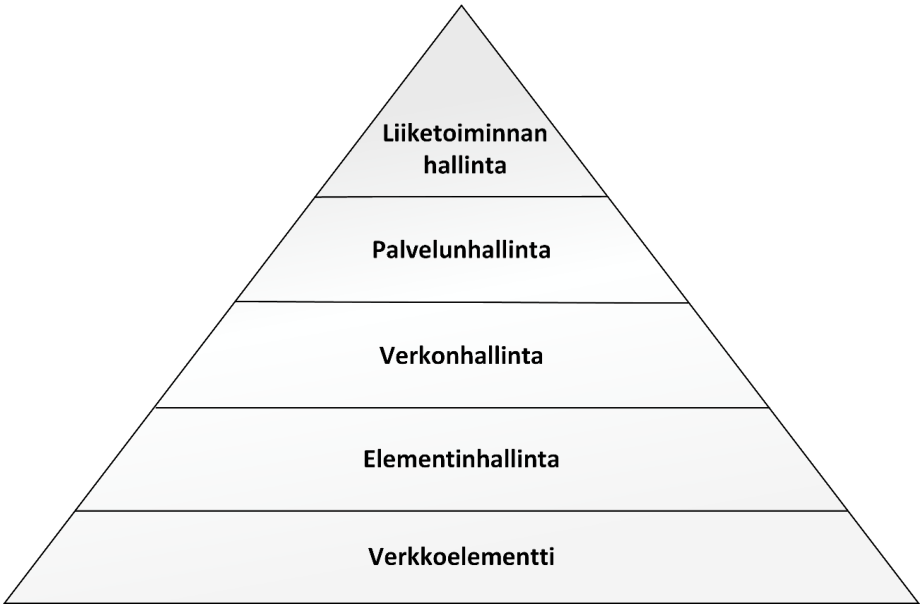
Toinen vaihtoehto FCAPS-hallintatoimintojen kategorisoinnille tunnetaan nimellä OAM&P. OAM&P-malli jaetaan neljään kategoriaan, joita ovat toiminta (operations), hallinta (administration), ylläpito (maintenance) sekä hankinta (provisioning). Mallin käyttö on suosittua erityisesti suurten telekommunikaatiopalveluiden tarjoajien keskuudessa. FCAPS-viitekehys on yleisempi muiden yritysten keskuudessa. [31]

4.1.1 TMN

Verkonhallinta voidaan jäsentää hierarkkisiksi kerroksiksi, jotka rakentuvat toistensa päälle. Yhdellä kerroksella keskitytään yksittäisten verkkolaitteiden hallintaan, kuten ohjelmistopäivityksiin ja laitteen oikeanlaisen toiminnan valvomiseen. Toisella kerroksella huolehditaan palveluiden hallinnasta, kuten verkkoresurssien jakamisesta palveluille. Vaikka kummassakin tapauksessa verkkoa hallitaan, hallintatoiminnot ja niiden

toteutustavat ovat erilaisia. Hierarkkinen malli rakentuu alhaalta ylöspäin, alkaen yksittäisten laitteiden hallinnasta ja siirryttäessä ylemmille kerroksille lähestytään verkon tukeman yrityksen liiketoiminnan hallintaa. Vakiintunut luokittelu verkonhallinnan hallintakerroksille on TMN-hierarkiamalli. [31]

TMN-viitekehys viittaa joukkoon ITU-T:n (International Telecommunications Union – Telecommunication Standardization Sector) tuottamia tietoliikenneverkkojen hallintaan liittyviä standardeja. TMN-viitekehys kattaa useita osa-alueita, jotka liittyvät hallintaverkkojen rakentamiseen. Periaatteet vaihtelevat riippuen siitä, minkälaista verkkoa ollaan rakentamassa ja minkälaisia vaatimuksia verkolla on. [31] Yksi monista TMN-viitekehyksen määrittelemistä osa-alueista on kuvassa 10 esitetty hierarkkinen kerrosmalli, joka koostuu viidestä hallintakerroksesta. Kerrokset alhaalta ylöspäin ovat verkkoelementti (network element), elementinhallinta (element management), verkonhallinta (network management), palvelunhallinta (service management), sekä liiketoiminnanhallinta (business management). [31, 35] Seuraavaksi esitellään lyhyesti jokaisen kerroksen tehtävät.



Kuva 10. *TMN-viitekehyksen hierarkkisen viitemallin määrittelemät hallintakerrokset. Mukailtu lähteistä [31, 35].*

Verkkoelementti

Verkkoelementtikerros on tärkeä osa tehokasta hallintajärjestelmää [31]. Verkkoelementit vastaavat verkon fyysisiä laitteita, joita hallitaan sekä valvotaan [31, 35]. Verkkoelementti on osa hallintatoimintoja ja se määrittelee mitä hallinta- ja valvontamenetelmiä kyseisen verkkolaiteen kanssa voidaan käyttää. Verkkoelementtikerros on perusta hallintahierarkialle ja kaikki muut hallintatoiminnallisuudet rakentuvat tämän kerroksen päälle. [31]

Elementinhallinta

Verkkoelementtien hallintakerros kattaa verkon yksittäisten laitteiden hallinnan ja toimintakyvyn ylläpitämisen. Kerroksen toiminnallisuuteen kuuluu verkkoelementtien konfiguraation hallinta sekä elementtien valvonta. Hallinta mahdollistaa elementtien konfiguraation seurannan sekä muokkaamisen. Valvonnan avulla voidaan kerätä elementtien tilatietoja, lokitietoja sekä hälytyksiä. [31, 35]

Verkonhallinta

Kolmas kerros viitekehyksessä on verkonhallinta. Tämän kerroksen vastuulla on verkkoelementtien välisten suhteiden ja riippuvuuksien hallinta. [31, 35] Verkonhallintakerros huolehtii verkkokokonaisuudesta. Sen avulla hallitaan verkkoyhteyksiä päästä päähän. Elementtienhallinta mahdollistaa yksittäisten verkkolaitteiden hallinnan, mutta se ei kata toimintoja, joiden avulla varmistetaan verkkokokonaisuuden eheys. Esimerkiksi aliverkkojen välinen yhteys vaatii, että kaikki siihen liittyvät verkkoelementit ovat konfiguroitu oikein kyseistä tarkoitusta varten. [31] Tämän kerroksen valvontatehtävät keskittyvät liikenteen, viiveiden sekä kapasiteetin seurantaa. [31, 35]

Palvelunhallinta

Palvelunhallinta vastaa verkon avulla tuotettavien ja jaettavien palveluiden hallinnasta sekä niiden toiminnan varmistamisesta [31, 35]. Palvelu on tuote, joka tuotetaan asiakkaalle ja jonka käytöstä asiakas maksaa. Tämä kerros toimii myös rajapintana asiakkaiden suuntaan. Kerroksen tehtävänä on palveluiden tarjoamisen lisäksi käyttäjien hallinta, palveluiden laadunhallinta sekä palveluiden suorituskyvyn valvonta. [35]

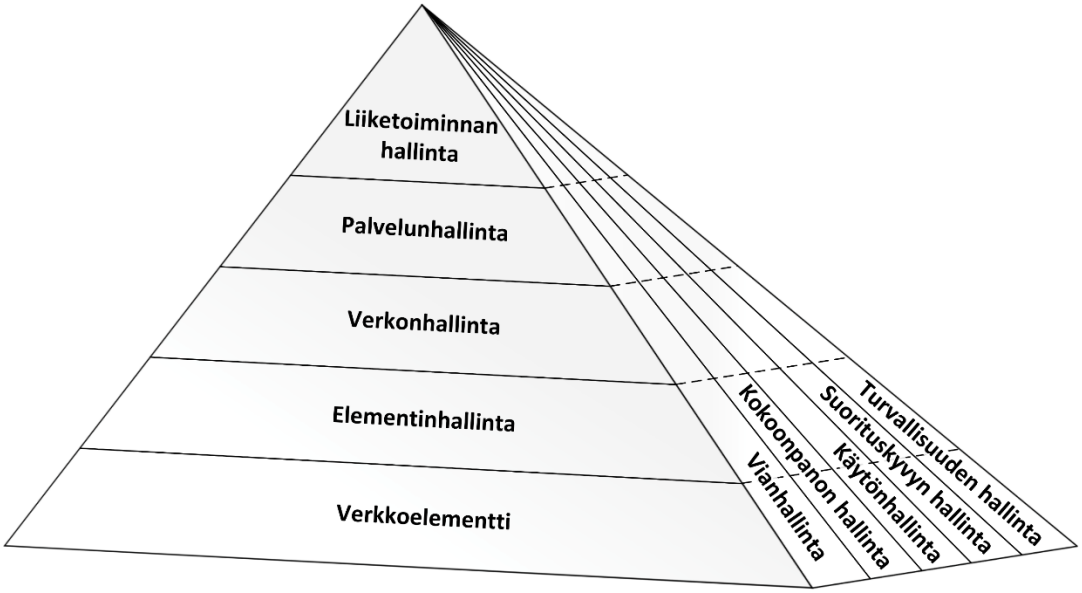
Liiketoiminnan hallinta

Liiketoiminnan hallinta liittää yrityksen liiketoiminnan ja edellä esitetyt tasot yhdeksi kokonaisuudeksi. Liiketoiminnan hallinta käsittelee palveluiden tuottamiseen liittyvän liiketoiminnan ja tarvittavien tukitoimintojen hallintaa. [31] Kerroksella on useita erilaisia tehtäviä. Esimerkiksi korkean tason suunnittelu, tulostavoitteiden määrittely, markkinatutkimukset, budjetointi sekä laskutus. [31, 35]

4.1.2 FCAPS

FCAPS on yleinen verkonhallinnan toimintamalli, joka on määritelty ISO:n ja ITU-T:n toimesta. Se kuvaa hallittavat alueet tietoliikenneverkoissa ja jakaa ne viiteen kategoriaan vianhallinta (fault management), kokoonpanon hallinta (configuration management), käytönhallinta (accounting management), suorituskyvyn hallinta (performance management) ja turvallisuuden hallinta (security management). [31, 33] Nämä viisi kategoriaa muodostavat verkonhallintajärjestelmien perustan [35]. Kategoriat käsitellään tarkemmin alapuolella. FCAPS-käsite on erittäin hyödyllinen ja tarjoaa yksinkertaisen

kehyksen verkonhallinnan toiminnallisten alueiden suunnitteluun. Se tarjoaa rakenteen hallintatoimintojen osa-alueille ja vahvistaa yhteisen terminologian. On kuitenkin huomattava, että FCAPS käsite sisältää yksinkertaistuksia. Useissa tapauksissa toiminnallisuuksia ei pystytä selkeästi kategorisoimaan, koska niiden käyttötarkoitukset vaihtelevat. [31] Kuvassa 11 on esitetty TMN-viitemallin ja FCAPS-toimintamallin välinen vertikaalinen integraatio.



Kuva 11. *FCAPS-toimintamallin integroituminen TMN-viitemallin kerroksille. Mukailtu lähteestä [31]*

Vianhallinta

Vianhallinta käsittelee verkossa ilmeneviä ongelmia, kuten laitteistoon, ohjelmistoon tai kommunikointipalveluihin liittyviä häiriöitä. Vianhallinnan tehtävä on valvoa verkon toimintaa ja reagoida normaalista toiminnasta poikkeaviin tapahtumiin. [31, 33] Vianhallinta sisältää ongelmien tunnistamisen, eristämisen, määrittämisen sekä ratkaisemisen verkon toimintakyvyn palauttamiseksi [33, 36, 37]. Tehokkaan vianhallinnan toteuttaminen edellyttää verkon laitteiden tilatietojen, tapahtumien ja hälytyksien seurantaa sekä näiden historiatietojen keräämistä [31, 33, 36]. Vianhallinnan avulla voidaan varmistaa verkon häiriötön käyttö, tunnistaa häiriöön johtavat ilmiöt enne varsinaista verkon lamaannuttavaa ongelmaa sekä pitää häiriöiden vaikutukset mahdollisimman vähäisinä ja lyhytkestoisina. Häiriötilanteessa voidaan suorittaa automaattisia korjaustoimenpiteitä tai generoida vikailmoitus verkon ylläpitäjälle. Verkon toimintakyky pyritään palauttamaan mahdollisimman nopeasti, mutta se saattaa edellyttää verkon uudelleen konfigurointia. Verkon konfigurointi on osa kokoonpanon hallintaa. [36] Alaluvussa 4.5 esitellään teknologioita, jotka soveltuvat osaksi vianhallintaa. Nämä teknologiat ovat SNMP, ICMP sekä Syslog (System Logging Protocol).

Kokoonpanon hallinta

Kokoonpanon hallinta käsittelee verkkolaitteiden asetuksien ja konfiguraatioiden muutoksia verkon saattamiseksi toiminnalliseen tilaan [31, 36, 37]. Kokoonpanon hallinnan avulla voidaan muokata verkon fyysistä ja loogista kokoonpanoa, varmistua verkon todellisesta kokoonpanosta sekä ylläpitää varmuuskopioita verkkolaitteiden asetuksista [31, 33, 37]. Verkon kokoonpanon hallinta voi yksinkertaisimmillaan olla yhden verkkolaitteen rajapinnan asetuksen muokkaamista, mutta verkkolaitteiden määrän kasvaessa yhdelle laitteelle tehtävä muutos voi vaatia muutoksia myös useisiin muihin verkkolaitteisiin. Tällaisessa tilanteessa kokoonpanon hallintatyökalujen merkitys korostuu. Verkon muut hallintatoimet ovat riippuvaisia kokoonpanon hallinnasta. Esimerkiksi vianhallinnassa on hankalaa diagnosoida verkon ongelmaa, jos verkon fyysinen ja looginen kokoonpano ei ole tarkasti tiedossa. [31]

Käytönhallinta

Käytönhallinta tarjoaa menetelmät tietoliikennepalveluiden käytön seurantaa, hallintaa sekä palveluiden käytön laskuttamiseen [31, 36, 37]. Käytönhallinta keskittyy toimintoihin, joiden avulla organisaatiot voivat tehdä liikevaihtoa ja saavuttaa tuottoa tarjoamiensa viestintäpalveluiden avulla. Organisaatio voi tarjota palveluita asiakkaille tai käyttää niitä sisäisesti. Riippumatta tarkoituksesta, tuotetut todelliset palvelut ja niiden kulutus on pystyttävä mittamaan. Mittaamalla voidaan arvioida palveluiden kulujen ja hyötyjen välistä suhdetta, hallita kustannuksia sekä laskuttaa palveluiden käyttäjiä. [31]

Suorituskyvyn hallinta

Suorituskyvyn hallinta käsittelee verkon suorituskyvyn tunnuslukujen mittaamista sekä verkon hienosäätöä suorituskyvyn parantamiseksi. Tunnusluvut vaihtelevat verkon kerroksesta riippuen, mutta mitattavia tunnuslukuja ovat esimerkiksi läpäisykyky, viiveet sekä laatu. [31, 33, 36] Joidenkin mittareiden avulla voidaan tunnistaa verkossa alkavia ongelmia, mikä mahdollistaa näihin reagoimisen ennen verkon vakavampaa vikaantumista [31, 33]. Keräämällä suorituskykytietoja pidemmältä aikaväliltä, voidaan tunnistaa verkon suorituskykyä rajoittavat pisteet. Tunnistamalla nämä pisteet, voidaan verkon suorituskyvyn lisäämiseksi aloittaa tarvittavien muutostöiden suunnittelu. [31, 36] Yksi suosituimmista ja tehokkaimmista menetelmistä suorituskyvyn mittaamiseen on tietojen kerääminen virtausteknologioiden avulla. Alaluvussa 4.5.4 on esitelty tarkoitukseen soveltuvia teknologioita NetFlow, IPFIX (IP Flow Export protocol) sekä sFlow (sampled Flow).

Turvallisuuden hallinta

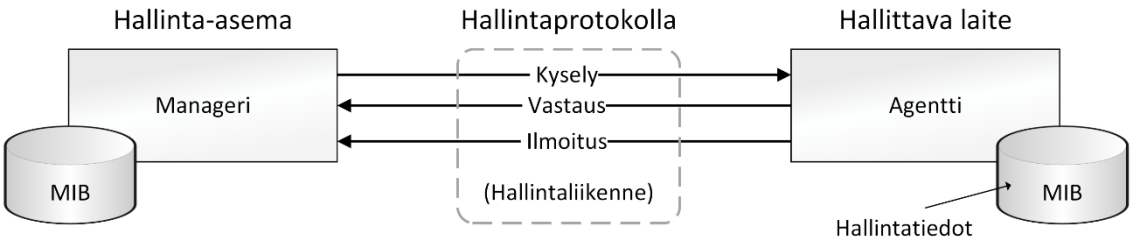
Turvallisuuden hallinta kattaa laaja-alaisesti erilaisia turvallisuuden näkökohtia. Turvallisuuden hallinta sisältää fyysisen tietoturvan, pääsynhallinnan sekä salauksen osa-alueet. [31, 33, 36] Turvallisuuden hallinta on syytä jakaa kahteen lähestymistapaan.

Hallinnan turvallisuuteen ja turvallisuuden hallintaan. Hallinnan turvallisuudella tarkoitetaan näkökulmia, joiden avulla toteutetaan turvallinen verkonhallinta. Tämä toteutetaan usein varmistamalla, että vain valtuutetuilla henkilöillä on pääsy hallintaverkkoon ja hallintatoimintoihin. Hallinnan suojaaminen pelkästään ulkopuolisia uhkia vastaan ei riitä, koska suojausrikkomukset voivat tapahtua myös organisaation sisältä. Turvallisuuden hallinnalla tarkoitetaan näkökulmia, joiden avulla toteutetaan verkkoinfrastruktuurin turvaaminen ulkoisia ja sisäisiä uhkia vastaan. [31] Turvallisuuden hallinta kattaa yrityksen resurssien ja yhteyksien suojaamista luvattomalta käytöltä. Salaus on yksi tärkeä osa-alue yrityksen verkkoliikenteen suojaamisessa. [36]

Hunajapurkit (honey pots) ovat esimerkki tekniikasta, joka avulla voidaan kerätä tietoa verkkoon kohdistuneista hyökkäyksistä ja verkon haavoittuvuuksista. Hunajapurkki näyttää hyökkääjän näkökulmasta laitteelta, joka on osa todellista tuotantoverkkoa. Todellisuudessa se on tuotantoverkosta eristetty ja hyvin suojattu laite, joka toimii ansana hyökkääjälle. Kaikkea hunajapurkkiin kohdistuvaa liikennettä voidaan pitää haitallisena. Liikenteen analysoinnin avulla tuotetaan tärkeää tietoa hyökkäyksistä ja kehitetään parempia suojausmekanismeja hyökkäyksiä vastaan. [31]

4.2 Verkonhallinnan arkkitehtuurit

Verkonhallinta-arkkitehtuurimallia kutsutaan ISO:n sekä IETF:n (Internet Engineering Task Force) viitekehyksissä manager/agent -arkkitehtuurimalliksi [18]. Lähes kaikki verkonhallinnan protokollat perustuvat manager/agent -arkkitehruutimallin mukaiseen toimintaan. Manager/agent -arkkitehtuurimalli on muokattu verkonhallinnan tarpeisiin asiakas/palvelin (client/server) -arkkitehtuurimallista. Manager/agent -mallin toiminta on päinvastainen kuin asiakas/palvelin -mallissa. Agentteja voi olla useita, mutta niiden toiminta vastaa asiakas/palvelin -mallin mukaisen palvelimen toimintaa. Managereita on yleensä yksi tai muutama ja ne muistuttavat toiminnaltaan asiakas/palvelin -mallin asiakasta. [32] Verkonhallinta-arkkitehtuurimalli sisältää neljä pääkomponenttia, jotka ovat esitetty kuvassa 12.



Kuva 12. *Verkonhallintaprotokollien toiminnan taustalla oleva manager/agent -arkkitehtuurimalli. Mukailtu lähteestä [18].*

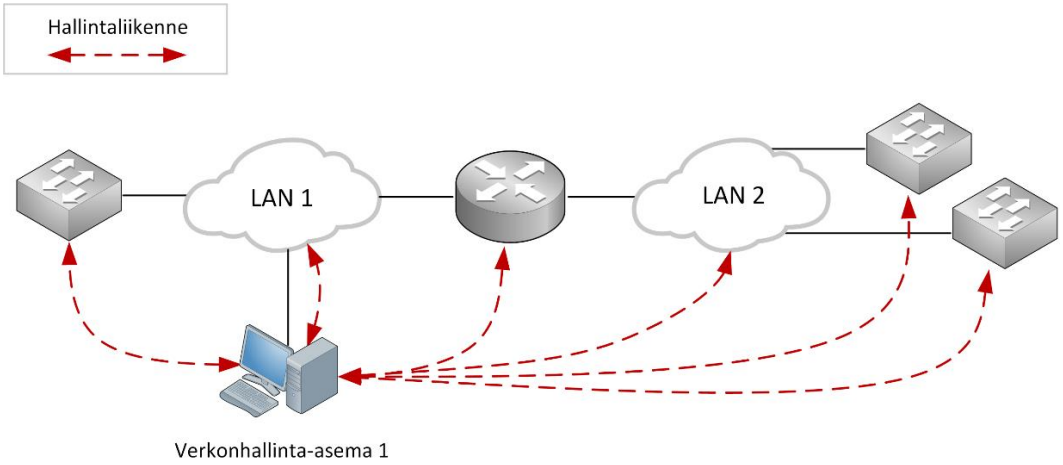
Verkonhallinta suoritetaan hallinta-asemalta. Hallinta-asema on tietokone, jossa suoritetaan erityisesti verkonhallintaan tarkoitettuja ohjelmistoja. Hallinta-asemat koostuvat

joukosta sovellusprosesseja, jotka suorittavat esimerkiksi verkkolaitteilta kerättyjen tietojen analysointia, verkon ongelmien tunnistamista sekä korjausta. Hallinta-aseman sisältämiä hallintasovelluksia kutsutaan manageреiksi (manager). Manageri tarjoaa hallintatietoja käyttäjille, lähettää kyselyt (request) hallittaville laitteille, vastaanottaa lähettämiensä kyselyiden vastaukset (response) sekä vastaanottaa hallittavien laitteiden tilaa koskevia raportteja. Näitä raportteja kutsutaan ilmoituksiksi (notification). Ilmoituksia käytetään ongelmien, poikkeavuuksien sekä hallittavien laitteiden tilan muutosten raportoimisessa managerille. [18, 31]

Hallittava laite on verkkolaite, joka tukee hallintaominaisuuksia. Hallittava laite voi olla esimerkiksi palvelin, reititin tai kytkin. Hallittavan laitteen sisältämää hallintaominaisuutta kutsutaan agentiksi (agent). Agentit kommunikoivat hallinta-aseman managerin kanssa ja suorittavat managerin pyytämät toimenpiteet hallittavassa laitteessa. Agentti käsittelee managerin pyynnöt, vastaa näihin pyyntöihin sekä tarjoaa itsenäisesti ilmoituksia managerille oman tilansa muutoksista. [18, 31] Hallittavat laitteet ylläpitävät yhtä tai useampaa hallintatietomuuttujaa, jotka kuvaavat laitteen tilaa. ISO:n ja IETF:n viitekehyksissä näitä muuttujia kutsutaan hallittaviksi objekteiksi (managed objects). [18] Näiden objektien kokoelmasta käytetään nimitystä MIB (Management Information Base) [18, 31]. Seuraavaksi esitellään kolme verkonhallinta-arkkitehtuuria, jotka tulee huomioida verkonhallintajärjestelmän suunnittelun yhteydessä. Verkonhallinta-arkkitehtuurit ovat keskitetty, hajautettu sekä hierarkkinen verkonhallinta.

Keskitetty verkonhallinta

Keskitetty verkonhallinta toteutetaan tyypillisesti yhdellä hallinta-asemalla, johon on integroitu kaikki verkonhallinnassa tarvittavat työkalut ja ohjelmistot. Keskitetyn verkonhallinnan arkkitehtuuri on yksinkertainen ja toteutuskustannuksiltaan edullisin tapa toteuttaa verkonhallintajärjestelmä. Kuvassa 13 on esitetty keskitetyn verkonhallinnan arkkitehtuuri, jossa kaikki verkonhallinta ja -valvonta toimenpiteet suoritetaan yhdestä keskitetystä pisteestä. Hallintatiedot kerätään, prosessoidaan sekä esitetään keskitetyn hallintajärjestelmän avulla. [37, 38]

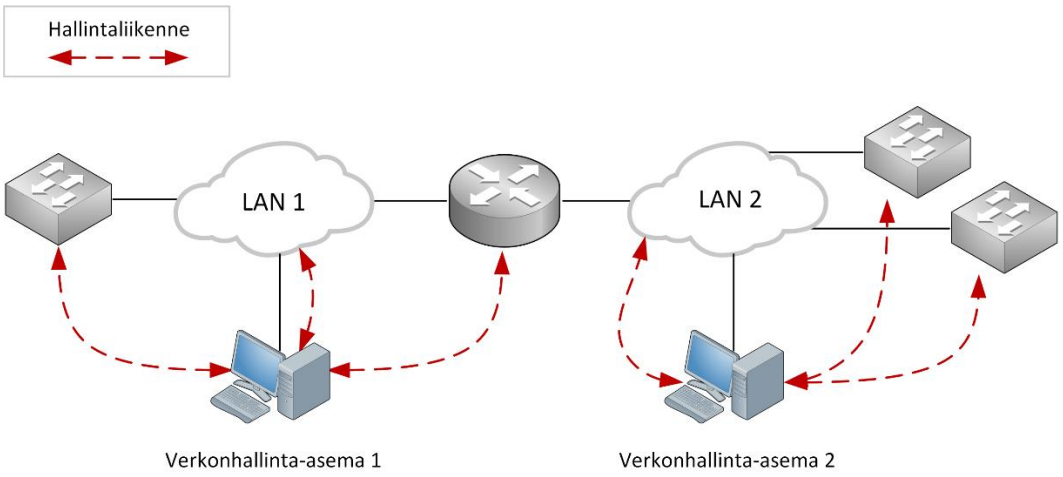


Kuva 13. *Keskitetyn verkonhallinnan arkkitehtuuri. Mukailtu lähteestä [37].*

Verkonhallinnan tapahtuessa yhden keskitetyn hallintajärjestelmän avulla, järjestelmän vikaantuminen voi pahimmassa tapauksessa lamaannuttaa koko verkonhallintatoiminnan. Esimerkiksi hallittavien laitteiden määrän kasvaessa, myös verkon hallintaliikenteen määrä kasvaa. Verkkoliikenteen määrä voi muodostua yhdellä hallinta-asemalla toteutetun verkonhallintajärjestelmän pullonkaulaksi. [37, 39]

Hajautettu verkonhallinta

Hajautettu verkonhallinta toteutetaan sijoittamalla useampia itsenäisiä hallinta-asemia verkon strategisiin segmentteihin [37-39]. Hallintaliikenne jakautuu paikallisille segmenteille, eikä tästä syystä kuormita vain yhtä verkonhallinta-asemaa. Kuvassa 14 näkyy useita paikallisia verkonhallinta-asemia sijoitettuna strategisiin segmentteihin. Hajautetussa verkonhallintajärjestelmässä voi olla useita itsenäisiä verkonhallinta-asemia tai yski hallinta-asema ja useita hallintatiedon kerääjiä. [37]



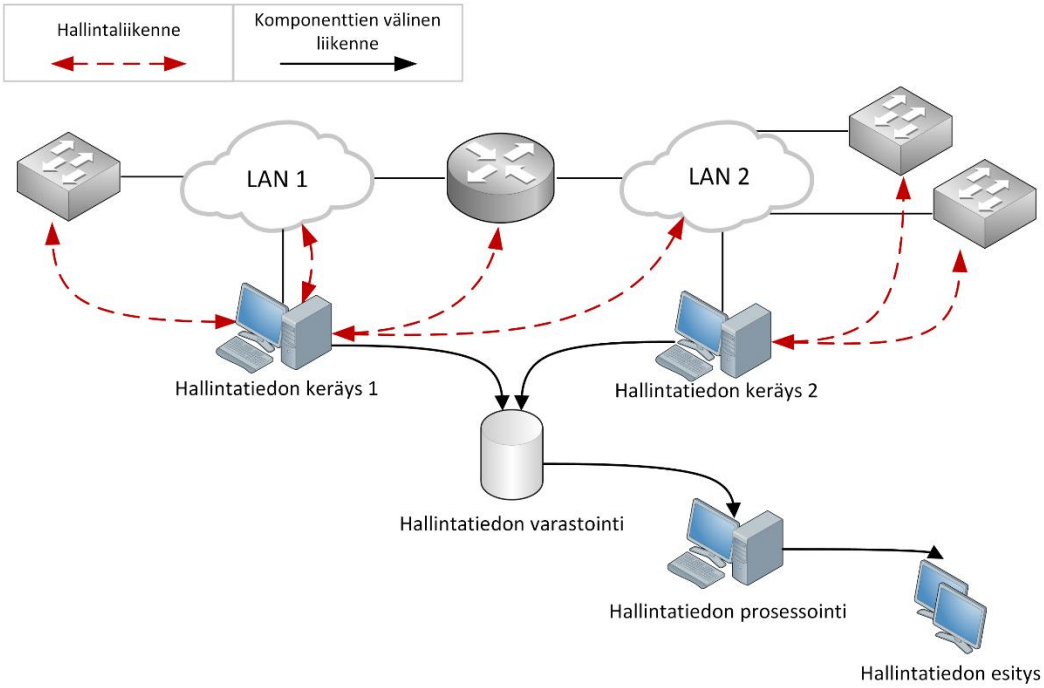
Kuva 14. *Hajautetun verkonhallinnan arkkitehtuuri. Mukailtu lähteestä [37].*

Hajautuksen avulla voidaan vähentää verkossa kulkevan hallintatiedon määrää sekä tarjota redundanttisuutta. Hallinta-asema voidaan konfiguroida korvaamaan toisesta seg-

mentistä kadonneen hallinta-aseman. Hajautetussa verkonhallinnassa useampien hallinta-asemien tarve kasvattaa kustannuksia, mikä on huomioitava arkkitehtuurin valintaa suunniteltaessa. [37]

Hierarkkinen verkonhallinta

Hierarkkinen verkonhallinta toteutetaan erottamalla hallintatoiminnot erillisiksi komponenteiksi. Hallintatiedon kerääminen, prosessointi, varastointi ja esittäminen voidaan toteuttaa erillisillä laitteilla. [37, 38] Kuvassa 15 hallintatoimet on eroteltu komponenteittain, jotka yhteystyössä muodostavat hierarkkisen verkonhallintajärjestelmän. Hierarkkisessa verkonhallinnassa paikalliset keräyslaitteet lähettävät hallintatiedot joko suoraan varastointi ja esityslaitteille tai käsittelylaitteille prosessoitavaksi. Kun hallintatiedot siirretään suoraan esitys- ja varastointilaitteisiin ilman prosessointia, hallintalaitteet toimivat samalla periaatteella kuin hajautetussa verkonhallinnassa. [37]



Kuva 15. *Hierarkkisen verkonhallinnan arkkitehtuuri. Mukailtu lähteestä [37].*

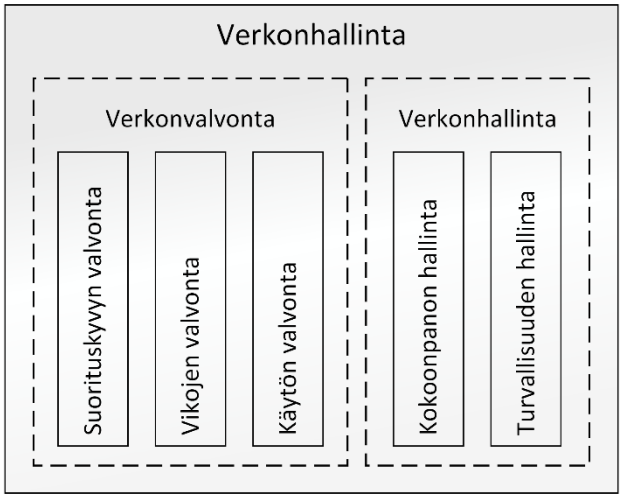
Hallintatietojen prosessointi, ennen tietojen siirtämistä esitys- ja varastointilaitteille, tekee keräyslaitteista eräänlaisia suodattimia. Keräyslaitteilla suoritettavan hallintatiedon prosessoinnin avulla voidaan siirtää vain tarpeellinen hallintatieto esitys- ja varastointilaitteille. [37] Suodattamalla ylimääräinen hallintaliikenne pois, voidaan merkittävästi vähentää hallintaliikenteen määrää verkossa [37, 39].

Hierarkkisen verkonhallinnan etuna on mahdollisuus toteuttaa komponentit redundanttisina ja toisistaan riippumattomina. Joissain verkonvalvontajärjestelmissä on tarve useille esityslaitteille ja toisissa järjestelmissä voidaan tarvita useita prosessointi- tai varastointilaitteita. Koska kaikki komponentit ovat erillisiä, niiden lukumäärät voidaan mää-

ritellä tarpeen mukaan. Hierarkkinen verkonhallintajärjestelmä muuttuu nopeasti monimutkaiseksi kokonaisuudeksi, joka on kallis toteuttaa sekä ylläpitää. [37]

4.3 Verkonvalvonta

Verkonhallinnan FCAPS-toimintamallin mukaiset osa-alueet voidaan jakaa verkonhallintaan sekä verkonvalvontaan. Verkonhallinnan tehtävänä on muokata verkon laitteiden konfiguraatiota sekä asetuksia. Verkonvalvonnan tehtävänä on tarkkailla verkon tilaa, konfiguraatiota sekä analysoida verkon suorituskykyä koskevia tietoja. Kuvassa 16 on esitetty osa-alueiden karkea jako hallintaan ja valvontaan. Osa-alueista koostuvaa kokonaisuutta kutsutaan verkonhallinnaksi. [3]



Kuva 16. *Verkonhallinnan osa-alueiden karkea jako verkonvalvontaan ja -hallintaan. Mukailtu lähteestä [3].*

Kuvassa 16 esitetyssä jaossa verkonvalvonta koostuu suorituskyvyn, vikojen sekä käytön valvonnasta. Tämä työ keskittyy verkonvalvonnan osa-alueisiin. Näistä osa-alueista tarkastellaan erityisesti suorituskyvyn sekä vikojen valvontaa.

Suorituskyvyn valvonta on verkon tilastotietojen keräämistä suorituskyvyn arvioimiseksi sekä verkon hienosäätämiseksi. Tavoitteena on mahdollistaa resurssien asianmukainen kohdentaminen verkkoon. Esimerkiksi pullonkaulojen poistaminen, ennusteiden antaminen verkon suunnittelua varten sekä tarjota näin paras mahdollinen palvelunlaatu. [31]

Vikojen valvonta koostuu valvontatoiminnoista, joiden avulla varmistutaan, että kaikki verkon laitteet ja palvelut toimivat oikein. Hälytysten ja jatkuvasti syntyvän suuren tapahtumamäärän käsittely ovat suuria haasteita vikojen valvonnassa. Vikojen valvonta sisältää myös vianetsintää sekä vikadiagnosointia. [31]

Käytönvalvonta koostuu verkon sekä palveluiden käyttöä koskevien tietojen keräämisestä ja tallentamisesta. Käytönvalvonta on keskeisessä asemassa, kun yrityksen tulos

tehdään tuottamalla verkkopalveluita. Käytönvalvonnan avulla pystytään määrittämään verkon avulla tuotettu arvo. [31]

Verkonvalvonnan perustehtävä on hälytysten kerääminen. Hälytykset ilmaisevat normaalista toiminnasta poikkeavista tapahtumista. Esimerkiksi hälytyksiä voivat olla verkkolaitteen rajapinnan vika, langattoman yhteyden signaalin laadun heikentyminen, kirjautumien verkkolaitteeseen tai epäilty tunkeutumisyritys verkkoon luvattoman käyttäjän toimesta. Hälytykset kerätään verkonhallintajärjestelmällä ja sovellus tai verkonylläpitäjä päättää, mitkä ovat hälytyksestä seuraavat toimenpiteet. Hälytyshistorian keräämisellä saavutetaan monia etuja. Hälytyshistorian avulla pyritään tunnistamaan hälytyksissä toistuvat sarjat. Toistuvuuksien tunnistamisen avulla on mahdollista löytää vian juurisyyt. Hälytyshistorian avulla voidaan arvioida myös verkon muutostöiden vaikutusta verkon toimintaan. [31]

Toinen tärkeä asia verkonvalvonnassa on verkon tilan visualisoiminen ylläpitäjälle. Visualisointi voidaan toteuttaa monilla tavoilla. Yksinkertaisimmillaan visualisointi on tekstiluettelo verkon tapahtumista. Informatiivisempi tapa toteuttaa visualisointi on käyttää topologiakarttoja. Kartan interaktiiviset kuvakkeet edustavat laitteita ja kuvaavat laitteiden tiloja. Laitteiden väliset yhteydet voidaan esittää kartalla esimerkiksi viivojen avulla. Topologiakartan avulla voidaan tarjota hyvä yleiskuva verkon kokonaistilasta ja osoittaa vikaantuneen verkkolaitteen maantieteellinen sijainti. Tämä helpottaa ongelman vianselvityksen aloittamisessa ja antaa vihjeen ongelman mahdollisesta sijainnista. [31]

4.4 Verkonvalvonnan vaatimukset

Tämän diplomityön keskittyessä verkkolaitteiden kunnonvalvontaan, tarkastellaan TMN-viitekehyksen kolmea alinta kerrosta (Kuva 11). Puhuttaessa verkon kunnonvalvonnasta FCAPS-toimintamallin käsittely rajataan vikojen valvontaa (F) ja suorituskyvyn valvontaa (P). Näiden osien avulla selvitetään verkon toimintakyky, tila ja tunnistetaan mahdolliset ongelmat. Seuraavaksi käydään läpi verkonvalvonnalle ja verkonvalvontaprotokollalle asetettuja vaatimuksia. Vaatimuksia tarkastellaan seuraavaksi vikojen valvonnan ja suorituskyvyn valvonnan näkökulmista.

Vikojen valvonta

Valvontatyökalujen käytön yksinkertaistamiseksi verkonvalvontaprotokollan tulisi tukea menetelmää, jonka avulla valvottavalta laitteelta voidaan kysyä, mitä ominaisuuksia se tarjoaa valvontarajapintaan. Usein valvottavaan verkkoon sisältyy suuri määrä laitteita. Valvonnan vaatimukset ja tarpeet muuttuvat ajan kuluessa, nämä muutokset voivat kasvattaa myös verkon laitteiden määrää. Tästä syystä verkonvalvontaprotokollan pitää olla skaalautuva laitemäärän sekä laitekohtaisten valvottavien objektien osalta. [40]

Verkkolaitteet sisältävät suuren määrän samantapaisia valvottavia objekteja, kuten laskureita. Useimmissa tapauksissa kaikki valvottavat objektit eivät ole merkityksellisiä verkonvalvonnan toteutuksen kannalta. On tärkeää pystyä valitsemaan tarvittavat objektit, koska valvontaoperaatiot pitää pystyä kohdistamaan haluttuihin valvontatietoihin. Valvontaprotokollalla pitää olla nimeämismekanismi, jonka avulla yksittäiset objektit voidaan tunnistaa. Valvontatoiminnot eivät myöskään saa vaikuttaa negatiivisesti verkkolaitteen ensisijaisiin tehtäviin. Verkonvalvontaprotokollalla tulee olla mahdollisimman vähäinen vaikutus valvottavan laitteen toimintaan sekä suorituskykyyn. [40]

Kyselypohjaisen valvonnan heikkoudet, kuten havaitsemattomat ongelmat kyselyiden välisenä ajankohtana, luo tarpeen tapahtumapohjaiselle valvontamekanismille. Tapahtumapohjaisen valvonnan avulla, verkkolaitteet voivat raportoida reaaliaikaisesti oman toiminnallisen tilan muutoksista, erityisesti virhetilanteista. Tällaisissa laitteen lähettämissä ilmoituksissa, tulee olla riittävästi tietoa tapahtuman lähteen tunnistamiseksi ja ajankohdan sekä vakavuusluokituksen määrittämiseksi. [40]

Tapahtumailmoituksien eheys sekä ilmoituksien lähettäjän aitous on kyettävä varmistamaan. Tämä on erityisen tärkeää tilanteissa, joissa ilmoituksien pohjalta käynnistetään automaattisia korjaustoimenpiteitä. Tapahtumailmoitusten luotettava toimitus tulee varmistaa esimerkiksi kuittausmenettelyllä. Luotettavuutta voidaan lisätä säilyttämällä tapahtumaloki myös verkkolaitteessa myöhempää tarkastelua varten. Tapahtumailmoituksia tulee pystyä tulkitsemaan koneellisesti sekä ihmisen ymmärtämässä muodossa. Ilmoituksen tulee sisältää koneellisesti tulkittava rakenteellinen muoto sekä selkokieliset tapahtumakuvaukset. [40]

Suorituskyvyn valvonta

On tärkeää kerätä verkon käyttöä kuvaavaa mittaustietoa. Tällaiset mittaustiedot ovat hyödyllisiä esimerkiksi verkon käyttöastetta seurattaessa, verkon suunnittelussa, palveluiden laadun seurannassa, vianselvityksessä sekä hyökkäysten ja tunkeutumisten valvonnassa. Mittaustietojen keräämiseen tarkoitettujen protokollien tulee olla skaalattavissa lukuisiin virtauksenseuranta prosesseihin. Lisäksi mittaustietoja pitää pystyä lähettämään useille verkonvalvonta-asemille. [40]

Verkkojen kaistanleveyksien kasvaessa ja valvontatehtäviin käytettävän prosessointinopeuden ollessa rajallista, on syytä huomioida mahdollinen tarve tilastollisille näytteenottotekniikoille. Mittaustietojen keräysprotokollan tulee olla luotettava, eikä se saa ylikuormittaa valvottavia verkkolaitteita. Järjestelmän ruuhkautuminen voi johtaa erilaisiin tietojen menetyksiin ja saattaa vaikuttaa verkon palveluihin negatiivisesti. Tietojen menetystä voi esiintyä myös mittaustietojen keruuprosessin aikana. Kokonaiskuvan aikaansaamiseksi mittaustietojen menetyksiä on pystyttävä seuraamaan. [40]

4.5 Verkonvalvontaan soveltuvat protokollat

Tässä alaluvussa esitellään verkonhallintaan ja verkonvalvontaan soveltuvia protokollia, jotka toimivat TCP/IP-viitemallin mukaisessa verkkoympäristössä. ICMP-protokollaa lukuun ottamatta kaikki tässä luvussa kuvatut protokollat toimivat aikaisemmin esiteltyjen OSI ja TCP/IP -viitemallien sovelluskerroksessa. Tässä luvussa esiteltävien protokollien lisäksi on olemassa lukuisia muita näihin tehtäviin soveltuvia protokollia.

Kaikille verkonhallinta-arkkitehtuureille on yhteistä hallittavien laitteiden ja verkonhallinta-aseman välisessä kommunikoinnissa käytettävät verkonhallintaprotokollat [38]. Verkonhallintasovellukset kommunikoivat hallittavien verkkolaitteiden sekä mahdollisesti toisten hallintasovellusten kanssa. Verkonhallintaprotokollat määrittelevät säännöt, joiden avulla järjestelmän kommunikointi tapahtuu. [31] Verkonhallintaprotokollan tulee huomioida tietoturvanäkökulmat suhteessa ympäristöön, jossa verkonhallintaa ollaan toteuttamassa. Protokollan tulee sisältää menetelmät tietojen salaamiseksi sekä käyttöoikeuksin todentamiseksi. Näin voidaan rajoittaa hallintatoimintoja suorittavien käyttäjien määrää sekä estää hallintatietojen joutumine ulkopuolisille tahoille. Verkonhallintaprotokollan tulee tarjota myös tehokkaita menetelmiä hallintatietojen reaaliaikaiseen seurantaan. Hallintatiedot pitää pystyä erottelemaan konfiguraatiotiedoiksi sekä laitteen tilatiedoiksi. [41]

4.5.1 Simple Network Management Protocol, SNMP

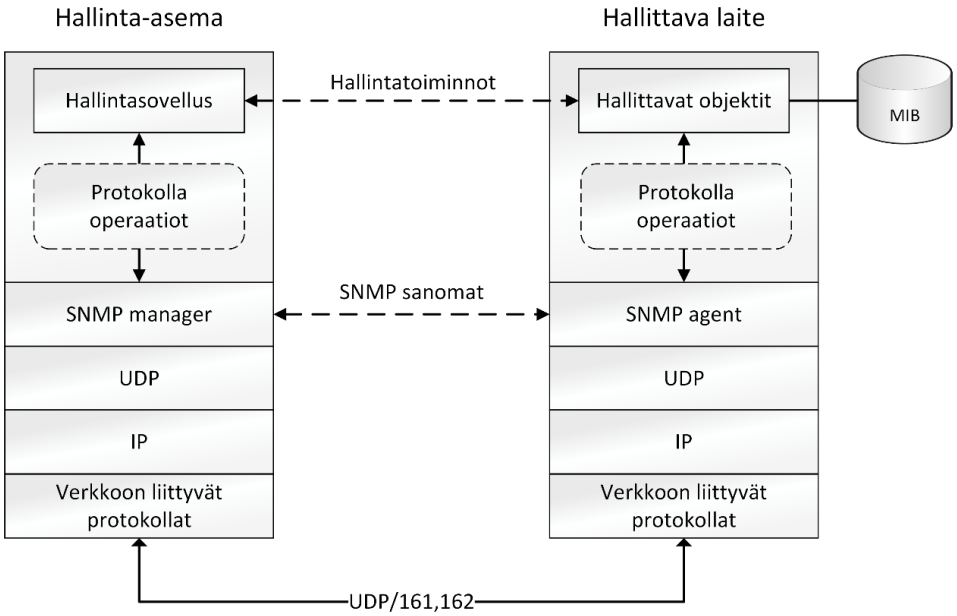
SNMP on IETF:n kehittämä verkonhallintaprotokolla, joka on yksi tunnetuimmista hallintaprotokollista. SNMP on kehitetty 1980-luvun lopulla ja se on saavuttanut vahvan aseman informaatioteknologian hallinta- sekä valvontaprotokollana. [31, 34, 42] SNMP on sovelluskerroksen protokolla, joka helpottaa hallinnoitavien verkkolaitteiden ja hallinta-aseman välistä tiedonvaihtoa [18, 43, 44]. SNMP toimii alaluvussa 4.2 kuvatun manager/agent -arkkitehtuurimallin mukaisesti ja on vakiintunut osa TCP/IP-protokollaperhettä [18, 43]. SNMP-protokollasta on olemassa kolme versiota SNMPv1, SNMPv2 sekä SNMPv3. Versiot rakentuvat toistensa päälle ja vaikka uusin versio on ollut käytettävissä pitkään SNMP:n ensimmäistä versiota käytetään edelleen lukuisissa toteutuksissa. [31]

Verkkoon, jota hallitaan SNMP-protokollalla, kuuluu yleensä useita agentteja, yksi agentti jokaista hallittavaa laitetta kohden. SNMP-agentti on sovellus, jota suoritetaan hallittavassa laitteessa, ja joka ylläpitää laitteen kokoonpanoa, tilaa sekä laskureita kuvaavia muuttujia. Näiden muuttujien kokoelmaa kutsutaan hallittavien objektien tietovarastoksi, MIB. [43]

SNMP-ilmoituksia hallittavan laitteen tilanmuutoksista kutsutaan, SNMP-versiosta riippuen, Trap tai Inform -sanomiksi. Näillä sanomilla on keskenään täysin samanlainen tarkoitus, mutta ne eroavat hieman toimintamekanismeiltaan. SNMP:n ensimmäisessä

versiossa määritellyn Trap-sanoman mekanismi on niin sanottu ”lähetä ja unohda”, jossa agentti lähettää Trap-sanomat hallinta-aseman IP-osoitteeseen UDP:n välityksellä. UDP-protokollalla toteutetussa sanomanvälityksessä, ilman sovellustason virheenkorjausta, on sanoman katoamisen riski. Inform-sanoma on määritelty SNMP:n toisessa versiossa ja se on samanlainen kuin Trap-sanoma, mutta sitä on kehitetty lisäämällä sanomanvälityksen luotettavuutta sovelluskerroksessa. Inform-sanoman kuljetuskerroksen protokollana toimii edelleen UDP-protokolla, mutta luotettavuutta on kasvatettu lisäämällä managerille velvollisuus kuitata vastaanotettu Inform-sanoma. Jos manageri ei kuittaa sanomaa vastaanotetuksi tietyn aikaikkunan sisällä, agentti lähettää sanoman uudelleen. Trap-sanoman mekanismi on kevyempi, kun taas Inform-mekanismi tarjoaa luotettavuutta, mutta aiheuttaa samalla kuormitusta agentille. Molemmat versiot sanomista ovat yleisesti käytössä tänä päivänä. [43]

SNMP hyödyntää kuljetuskerroksen UDP-protokollaa managerin ja agentin välisessä sanomanvaihdossa. UDP-protokollan yhteydettömän luonteen takia verkon suorituskykyä kuormittava vaikutus vähenee. SNMP-protokollasta on olemassa toteutuksia, jotka hyödyntävät kuljetuskerroksen TCP-protokollaa. TCP-pohjaiset toteutuksen ovat kuitenkin erikoistapauksia varten. SNMP-protokolla on suunniteltu toimimaan mahdollisimman hyvin ruuhkautuneessa ja vikaantuneessa verkossa, jolloin UDP-protokolla on ilmeinen valinta. UDP-protokolla on ruuhkautuneen verkon kannalta parempi suunnitteluvalinta kuin TCP-protokolla, joka saattaa aiheuttaa lisää ongelmia uudelleenlähetyksillä pyrkiessään saavuttamaan paremman luotettavuuden. [34] Kuvassa 17 on esitetty SNMP-verkonhallintaprotokollan arkkitehtuuri sekä SNMP-agentin ja managerin väliset riippuvuudet.



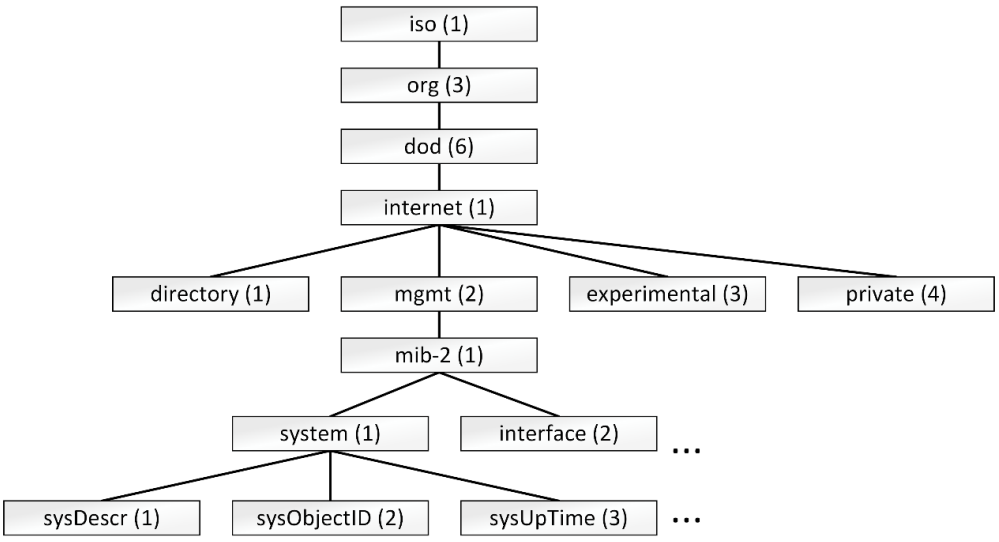
Kuva 17. *SNMP-protokollan arkkitehtuuri sekä agentin ja managerin välinen riippuvuus. Mukailtu lähteestä [18].*

SNMP-protokollan ensimmäisen version ongelmat kohdistuvat tietoturvaan ja protokollaoperaatioiden tehottomuuteen useiden satojen hallittavien laitteiden verkoissa. SNMP-protokollan toisessa versiossa korjattiin hallintaoperaatioihin kohdistuvat puutteet ja lisättiin operaatioita hallintatoimien tehostamiseksi. SNMP:n toinen versio ei kuitenkaan korjannut merkittävästi tietoturvaan liittyviä riskitekijöitä. Vasta SNMP-protokollan kolmas versio tuo korjaukset puutteellisiin tietoturvanäkökohtiin pitäen protokollamäärittelyn muuten ennallaan. Kolmas versio mahdollistaa hallintasanomien salauksen sekä managereiden ja agenttien vahvan todentamisen. SNMP-protokollan kolmas versio on huomattavasti aikaisempia versioita tehokkaampi, mutta samalla myös monimutkaisempi. [31]

Management Information Base, MIB

MIB on hallittavien objektien tietovarasto. Se määrittelee kaikki ne agentin hallinnassa olevat objektit, joita voidaan managerin avulla seurata sekä muokata. Objektit sisältävät laitteen tilaa, kokoonpanoa ja statistiikkaa koskevat tiedot. [18, 34, 43]

MIB-tietovarastossa sijaitsevat objektit tunnistetaan yksilöllisen OID (Object Identifier) -yksilöintitunnuksen avulla. Useimmissa laitteissa MIB-tietovarasto järjestää yksilöintitunnukset hierarkisesti IETF-standardeihin perustuen. Näiden standardoitujen yksilöintitunnusten lisäksi on olemassa valmistajien laitekohtaiset objektit sekä niiden yksilöintitunnukset. Laitekohtaiset yksilöintitunnukset vaihtelevat laitteen valmistajasta riippuen. [34, 43] Valmistajakohtaisia OID-yksilöintitunnuksia hallinnoi IANA (Internet Assigned Numbers Authority) -järjestö. Laitevalmistajat voivat vapaasti lisätä objekteja oman yksilöintitunnuksensa alle. Valmistajakohtaiset yksilöintitunnukset sijoittuvat MIB-hierarkian private-haaraan. [34] Kuvassa 18 on esitetty MIB-tietovaraston hierarkkinen puurakenne. Esimerkiksi järjestelmän toiminta-aikaa kuvaava objekti löytyy puurakenteesta OID-yksilöintitunnuksella 1.3.6.1.2.1.1.3.



Kuva 18. *Hallintaobjektien hierarkkinen tietovarasto, MIB. Mukailtu lähteestä [34].*

SNMP-protokollan MIB-tietovaraston hallittavien objektien tietotyypit ja niiden nimeämiskäytännöt määritellään hallintatietojen rakenteen, SMI (Structure of Management Information), syntaksin avulla [18, 34]. Hallintatietojen rakenteen määrittelystä on olemassa kaksi versiota SMIv1 ja SMIv2. Ensimmäinen SMI-versio on alkuperäisen SNMP-protokollaversion objektien määrittelyssä käytetty syntaksi. SMIv2 tuo parannuksia ensimmäiseen versioon ja se on esitelty SNMP-protokollan toisen version yhteydessä. SMI:n toinen versio määrittelee esimerkiksi uusia tietotyyppejä. [34] SMI itsessään pohjautuu OSI:n määrittelemään ASN.1 (Abstract Syntax Notation 1) -rajapinnan kuvaus kieleen [18, 34, 45]. ASN.1 -notaation etuna on alustariippumattomuus, joka mahdollistaa sanomien vaihdon eri järjestelmien välillä [34].

4.5.2 Internet Control Message Protocol, ICMP

ICMP-protokollaa käytetään verkkoyhteyskerroksella laitteiden välisessä kommunikoinnissa [4]. Protokolla on integroitu osaksi IP-protokollaa ja on toteutettava jokaiseen IP-moduuliin [21, 46]. IP-protokolla ei itsessään ole luotettava ja ICMP-sanomien tarkoituksena on tarjota tietoa viestintäympäristöön liittyvistä ongelmista [46]. ICMP-sanomia ei käytetä pelkästään virhetilojen raportoimiseen [4]. Taulukossa 2 on listattu 11 yleisintä ICMP-protokollan viestityyppiä.

Taulukko 2. *ICMP-protokollan yleisimmät viestityypit. Mukailtu lähteistä [4, 46].*

Code	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

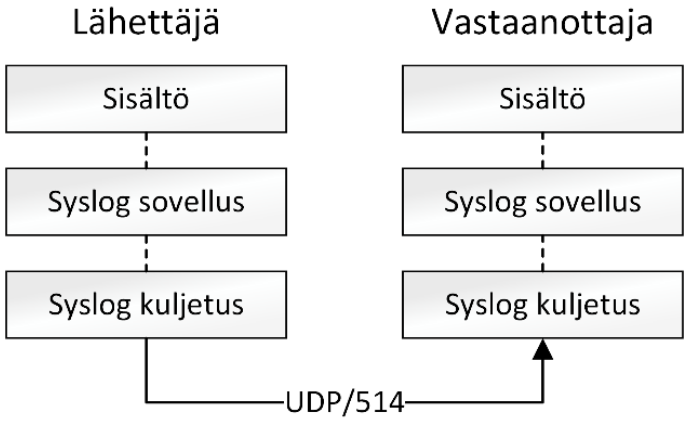
ICMP-protokolla mahdollistaa virheiden raportoinnin lisäksi erilaisten vianmääritystyökalujen toiminnan. Tällaisia ICMP-protokollaan perustuvia työkaluja ovat esimerkiksi PING (Packet Internet Groper) ja Traceroute. [21] PING käyttää hyväkseen ICMP-protokollan Echo-sanomia, jotka lähetetään halutulle kohteelle. Kohteen vastauksesta selviää kohteen tila sekä lähteen ja kohteen välinen viive verkossa. [20, 21] Traceroute-ohjelmalla voidaan selvittää verkon looginen reitti lähteestä kohteeseen. Tracerouten

avulla saadaan ICMP-virhesanomat kaikilta reitillä toiminnassa olevilta laitteilta. Tämän avulla voidaan selvittää esimerkiksi rikkoutuneen laitteen sijainti. [21]

4.5.3 System Logging Protocol, Syslog

Syslog on tiedonsiirtoprotokolla, jota käytetään tapahtumailmoitusten välittämiseen [47]. Syslog-protokollaa hyödynnetään järjestelmäsanomien keräämiseksi lokitiedostoon tai tietokantaan, jolloin syntyy yleiskuva laitteiden ja järjestelmän toiminnasta. Syslog-sanomat voivat sisältää kaikkea kriittisistä hälytyksistä yleisiin informatiivisiin ilmoituksiin. Lokitietojen jäljittäminen ongelmatilanteessa voi tuottaa korvaamattoman arvokasta tietoa vian juurisyistä. [31]

Syslog on kevyt ja yksinkertainen viestintäprotokolla, joka ei tue kaksisuuntaista liikennettä. Syslog on periaatteeltaan epäluotettava protokolla, joka ei tarjoa kuittausta tai varmistusta sanomien toimituksesta. [40, 47] Syslog-protokolla hyödyntää kerrosarkkitehtuuria, joka mahdollistaa erilaisten kuljetuskerroksen protokollien käytön Syslog-sanomien siirrossa [47]. Syslog-standardi määrittelee kolme kerrosta, joista jokaisella on tietty tehtävä protokollan toiminnan kannalta. Kuvassa 19 esitetyt kerrokset ovat Syslog sisältökerros (Syslog content), Syslog sovelluskerros (Syslog application) ja Syslog kuljetuskerros (Syslog transport). Syslog sisältökerros käsittää sanoman hyötykuorman. Syslog sovelluskerros huolehtii sanomien generoinnista, tulkinnasta, reitityksestä sekä tallennuksesta. Syslog kuljetuskerroksen tehtävä on sanomien välittäminen ja vastaanottaminen kuljetuskerroksen protokollalta. [40, 47] Perinteisesti Syslog-sanomien siirtoon on käytetty UDP-protokollaa [47] ja porttia 514 [48]. Syslog-sanomien siirtoon on mahdollista käyttää myös muita kuljetuskerroksen protokollia [47].



Kuva 19. *Syslog-protokollan arkkitehtuuri. Mukailtu lähteestä [47].*

4.5.4 Flow teknologiat

Yksi verkkoliikenteenvalvonnan passiivinen seurantatapa on pakettien kaappaus (packet capture). Tämä menetelmä tarjoaa yleensä eniten tietoa verkkoliikenteestä, koska paketit kaapataan kokonaisuudessaan myöhempää analysointia varten. Nopeissa verkoissa pakettien kaappaaminen vaatii kuitenkin erityistä laitteistoa ja huomattavaa infrastruktuuria pakettien tallennusta ja analysoimista varten. Huomattavasti paremmin skaalautuva passiivinen verkkoliikenteenvalvonnan menetelmä on virtauksenseuranta (flow expor). Virtauksenseurannassa verkkoliikenteen pakettien tiedot kerätään virtauksiin (flows). Virtauksen päätyttyä virtaustallenteet (flow records) lähetetään varastoitavaksi sekä analysoitavaksi. [49]

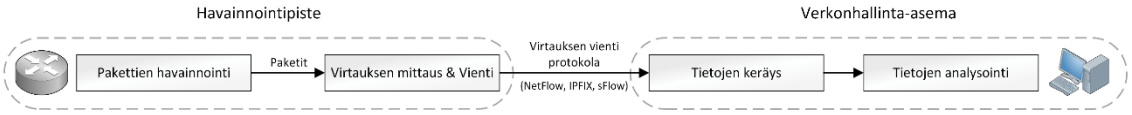
Termille virtaus on olemassa useita erilaisia määrittelyitä. RFC 3917 (Request For Comments) määrittelee virtauksen joukoksi IP-paketteja, jotka kulkevat verkon havainnointipisteen (observation point) lävitse tietyn ajanjakson sisällä. Kaikilla tiettyyn virtaukseen kuuluvilla paketeilla on joukko yhteisiä ominaisuuksia. [50] Nämä yhteiset ominaisuudet voivat sisältää esimerkiksi yhtenäiset pakettien otsikkokentät, kohde ja lähde IP-osoitteet sekä porttinumerot. [49, 50] Yksi virtaus koostuu kaikesta verkkoliikenteestä, joka kuuluu samaan viestintäkontekstiin. Käytännössä tällä tarkoitetaan IP-paketteja, jotka kuuluvat samaan istuntoon tai yhteyteen. [31] Termi virtaustallenne on määritelty RFC 7011:ssä. Virtaustallenne sisältää tiedot tietystä virtauksesta, jota on seurattu havainnointipisteessä. [51] Virtaustallenne sisältää virtauksesta mitatut ominaisuudet, esimerkiksi pakettien ja tavujen kokonaismäärän. [49, 51]

Virtauksenseurantateknologiat tarjoavat muutamia etuja pakettien kaappausteknologioihin verrattuna. Virtauksenseurantateknologiat ovat hyvin tuettuja ja laajasti integroituina erilaisiin paketinvälityslaitteisiin kuten reitittimiin, kytkimiin sekä palomuureihin. Verkkoinfrastruktuurin lisäksi ei tarvita erillisiä laitteita, minkä ansiosta virtauksenseuranta on edullisempi vaihtoehto kuin pakettien kaappaus. [49] Virtauksenseurantaa käytetään laajasti esimerkiksi tietoturva analyyseihin, kapasiteetin suunnitteluun, laskutukseen sekä profilointiin [49, 52, 53]. Viimeaikaisten tutkimusten lähestymistavat ovat keskittyneet pääasiassa verkkoturvallisuuden analysointiin. Tämän tarkoituksena on havaita verkossa tapahtuva poikkeava toiminta, jota perinteiset tietoturvainfrastruktuurit, kuten tunkeutumisen tunnistusjärjestelmät, IDS (Intrusion Detection System), palomuurit ja virustentorjuntatyökalut eivät pysty havaitsemaan. Nämä lähestymistavat käyttävät kuitenkin lisäksi kehittyneitä menetelmiä, kuten koneoppimista. [52]

Virtaustallenteet sisältävät korkean tason kuvauksen yhteyksistä, mutta eivät todellista siirrettyä tietoa [49, 52]. Tästä syystä virtauksenseuranta on tietoturvallisempaa kuin pakettien kaappaus. Vaikka virtauksenseuranta vähentää huomattavasti analysoitavan tiedon määrää verrattuna pakettien kaappaamiseen, voi tietokantojen koko kasvaa nopeasti kymmeniin teratavuihin. Tästä syystä virtauksenseurannan tuottamat tiedot olisi hy-

vä mieltää Big Datana ja tähän tarkoitukseen soveltuvien työkalujen avulla helpottaa tiedon keräämistä, käsittelyä sekä analysointia. [49]

Tyypillinen verkkoliikenteen virtauksenseurannan rakenne koostuu useista vaiheista, jotka ovat esitetty kuvassa 20. Ensimmäinen vaihe on pakettien havainnointi (packet observation), jossa paketit kaapataan havainnointipisteestä ja esikäsitellään [49, 52]. Havainnointipiste voi olla esimerkiksi paketinvälityslaitteen rajapinta [49]. Toinen vaihe on virtauksen mittaus ja vienti, joka koostuu mittausprosessista sekä vientiprosessista [49, 52]. Mittausprosessissa paketit yhdistetään virtauksiin ja kun virtauksen katsotaan päättyneen, virtaustallenne lähetetään virtauksenvientiprotokollan avulla tietojen keruulaitteille. Mittaus ja vientiprosessit ovat käytännössä läheisesti toisiinsa liittyviä. [49] Kolmas vaihe on tiedonkeruu [49, 52]. Sen tehtävä on edellisen vaiheen tuottamien virtaustietojen vastaanotto, varastointi ja esikäsittely. Nämä esikäsittelytoiminnot käsittävät tietojen yhdistämisen, suodatuksen, pakkaamisen sekä yhteenvedon muodostamisen. [49] Viimeinen vaihe on tietojen analysointi [49, 52]. Analyysitoiminnot sisältävät esimerkiksi liikenteen profiloinnin ja luokittelun, poikkeavuuksien ja häirinnän tunnistamisen, tiedon arkistoinnin sekä tiedon hakemisen esimerkiksi tutkimus tarkoituksiin. Toiminnallisessa ympäristössä tietojen keräys ja analysointivaiheet ovat usein yhdistettyjä toimintoja. [49]



Kuva 20. *Verkkoliikenteen virtauksenseurannan rakenne. Mukailtu lähteestä [49].*

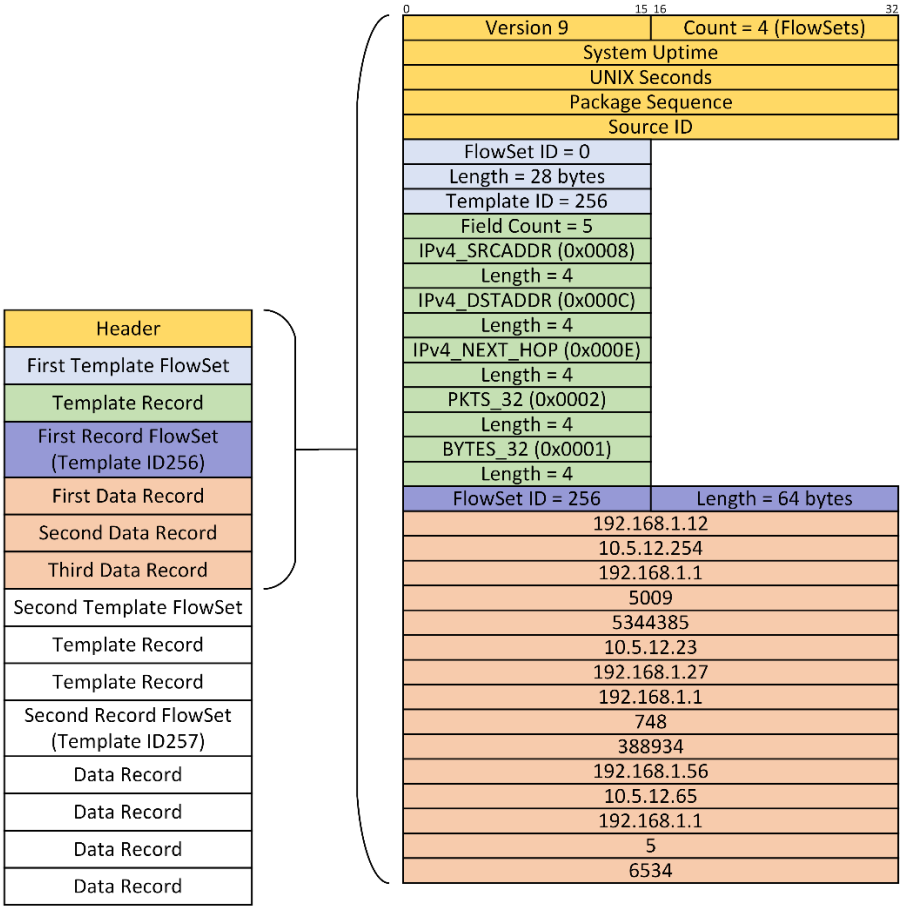
Termi virtaus viittaa useisiin teknologioihin, jotka pyrkivät ratkaisemaan keskenään samanlaisia ongelmia. [49] Seuraavaksi tarkastellaan muutamia yleisesti käytössä olevia sekä laajasti implementoituja teknologioita. Tarkasteltavat teknologiat ovat NetFlow, IPFIX sekä sFlow.

NetFlow

NetFlow on Cisco Systemsin kehittämä teknologia verkkoliikenteen seurantaan. Teknologia tarjoaa pääsyn IP-virtaustietoihin, jotka kulkevat havainnointipisteen lävitse. [31, 52, 54] NetFlow on useiden laitevalmistajien tukema protokolla, joka on sulautettu verkkolaitteisiin [52]. NetFlow keskittyy OSI-viitemallin kerrosten kolme ja neljä tietojen keräämiseen. NetFlow:n yhdeksäs versio laajentaa teknologian myös OSI-viitemallin toisen kerroksen tietojen keräämiseksi. [55]

Verkkolaitteet, joilta virtaustietoja kerätään, tarkastelevat liityntärajapintaan saapuvia paketteja ja kaappaavat virtaustiedot jokaisesta paketista tai perustuen näytteenotto ja suodatus määrittelyyn. Kerätyt virtaustiedot tallennetaan verkkolaitteen välimuistiin. Välimuistiin luodaan uusi virtaus ensimmäisestä paketista ja sitä päivitetään jatkuvasti

samoilla ominaisuuksilla saapuvien pakettien myötä. Välimuistiin kerätyt virtaustiedot lähetetään UDP:n tai SCTP:n (Steram Control Transport Protocol) välityksellä hallinta-asemalle. Virtaustietojen lähetys voi tapahtua jaksottaisesti ajastettuna tai virtausvälimuistin hallintaan perustuen. Näytteistys on vaihtoehto, jonka avulla voidaan pienentää kuormitusta vähentämällä käsiteltävien pakettien määrää. Näytteenotto voidaan konfiguroida toteutettavaksi tietyn pakettimäärän välein tai satunnaisesti vaihtuvin välein. NetFlow:sta on olemassa yhdeksän versioita, joista versiot viisi ja yhdeksän ovat yleisimmin käytettyjä. [52] Kuvassa 21 on esitetty NetFlow v9 virtaustallenne esimerkki.

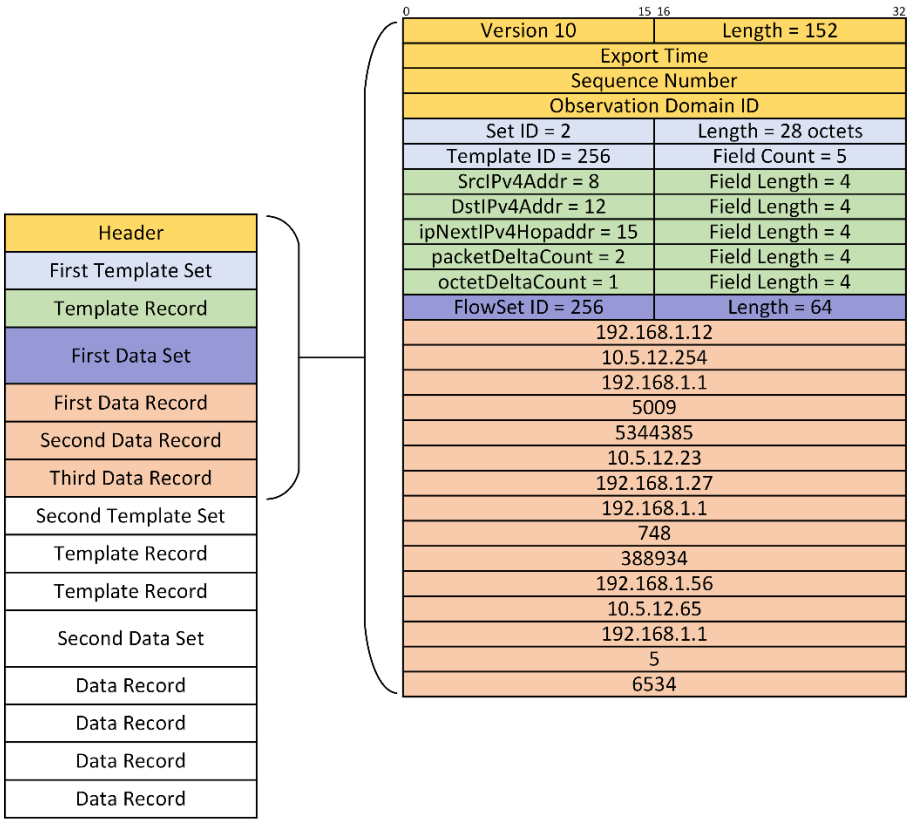


Kuva 21. *Esimerkki NetFlow versio 9 virtaustallenteesta. Mukailtu lähteestä [56].*

IP Flow Export protocol, IPFIX

IETF perusti työryhmän verkkoliikenteenseurantaan soveltuvan protokollan kehittämistä varten. Kehitystyö alkoi NetFlow-protokollan kehityksen kanssa päällekkäin ja IETF:n kehittämän protokollan nimeksi tuli IPFIX. Työryhmän tehtävä oli ensimmäiseksi määritellä protokollan vaatimukset ja arvioida useita ehdokasprotokollia. [49] Osana tätä arviointia NetFlow v9 valittiin IPFIX-protokollan perustaksi [49, 52]. IPFIX ei kuitenkaan ole pelkkä standardiversio Ciscon NetFlowsta, vaan se tuo mukanaan useita uusia ominaisuuksia [49]. IPFIX keskittyy NetFlown tavoin OSI-viitemallin kerrosten 2-4 virtaustietojen keräämiseen [55].

IPFIX on suunniteltu vastaamaan verkkoliikenteen valvonnan nopeasti kasvaviin vaatimuksiin tarjoamalla laajennettavan sekä joustavan tietomallin. Tietomalli voidaan räätälöidä vastaamaan erilaisia tarpeita. Protokolla tukee myös luotettavaa ja tietoturvallista tiedonsiirtoa SCTP, TCP sekä UDP -kuljetusprotokollia hyödyntäen. [52] IPFIX ja NetFlow -protokollan yhdeksäs versio ovat käytännössä hyvin samanlaiset. Teknisestä näkökulmasta katsottuna molemmilla protokollilla on sama tavoite ja samanlaiset yleiset periaatteet. Suurempi ero on poliittinen, IPFIX on avoin standardoimisorganisaation ylläpitämä protokolla, kun taas NetFlown määrittelystä vastaa yritys. [31] Kuvassa 22 on esimerkki IPFIX-protokollan virtaustallenteesta.



Kuva 22. *Esimerkki IPFIX virtaustallenteesta. Mukailtu lähteestä [51].*

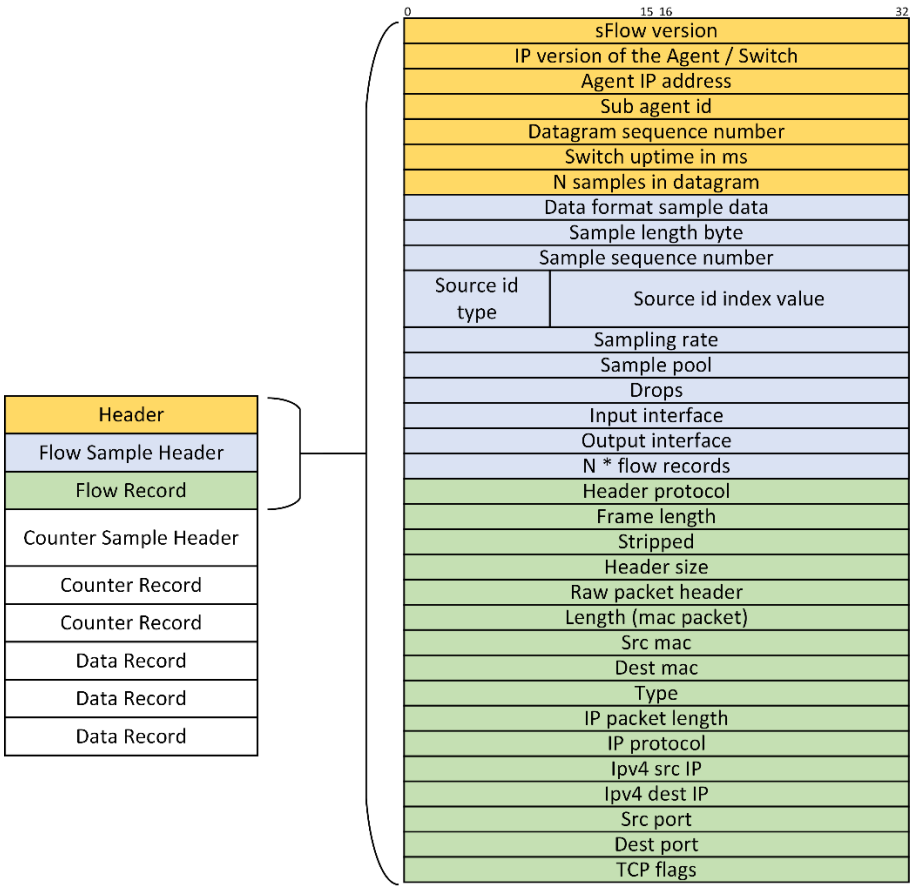
Sampled Flow Protocol, sFlow

Verkonvalvontaprotokolla sFlow on alun perin InMonin kehittämä [52, 57] ja saavuttanut laajan hyväksynnän useiden laitevalmistajien keskuudessa. sFlow on teollisuusstandardi nopeiden kytkentäisten ja reititettyjen verkkojen valvontaan. [49, 57, 58] Teknologia eroaa hieman aiemmin esitellyistä virtauksenseurantateknologioista. sFlown kyvyt pakettien tietopalojen sekä verkkolaiteiden rajapintalaskureiden keräämiseksi eivät ole tyypillisiä ominaisuuksia muille virtausteknologioille. sFlow ei tue 1:1 pakettinäytteenottoa, joka on oletusarvoista muille virtausteknologioille. Arkkitehtuuriltaan NetFlow, IPFIX sekä sFlow ovat kuitenkin hyvin samantapaiset. [49] sFlow on suunniteltu tarkkaan verkkoliikenteen seurantaan ja se skaalautuu kymmeniin tuhansiin valvottaviin kohteisiin yhdellä keräilijällä [58]. sFlow on kehitetty valvontateknologiaksi, joka tarjo-

aa hyvän skaalautuvuuden ja yksityiskohtaisen raportoinnin verkkoliikenteestä. sFlow-teknologia on suunniteltu OSI-viitemallin kerrosten 2-7 tietojen keräämiseksi. [52, 55, 57]

sFlow-valvontajärjestelmä koostuu agentista, joka on osa verkkolaitetta, sekä keskitetystä keräilijästä [44, 52, 58], joka on tavallisesti palvelimella suoritettava ohjelmisto [52]. sFlow-agentin tehtävä on koostaa verkkolaitteen liityntärajapintojen laskurit sekä pakettivirtojen näytteet ja lähettää ne sFlow-protokollalla keräilijälle UDP-kuljetuskerroksen protokollaa hyödyntäen [52, 58]. UDP-kuljetusprotokollan käyttö vähentää tietojen puskurointiin tarvittavan muistin määrää ja tarjoaa robustin tavan toimittaa tiedot oikea-aikaisesti myös verkon ollessa ruuhkautunut. UDP-kuljetusmekanismin epäluotettavuus ei vaikuta oleellisesti sFlown avulla toteutetun verkkoliikenteenvalvontajärjestelmän luotettavuuteen. [58] Virtausta ei säilytetä verkkolaitteen välimuistissa, edellä esiteltyjen protokollien tavoin, vaan näytteet lähetetään heti näytteenoton yhteydessä keräilijälle. Välittömästi tapahtuva tiedonsiirto minimoi verkkolaitteen muistin sekä suorittimen kuormituksen. [52]

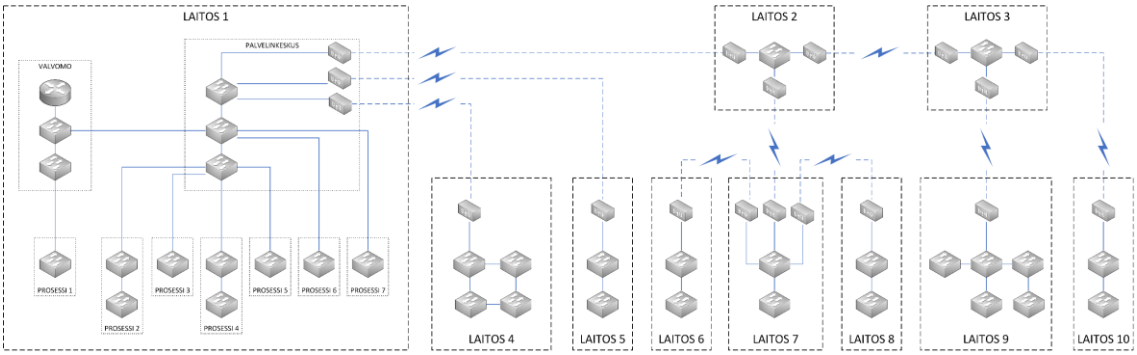
Näytteenottomekanismeja sFlow-agentilla on kaksi. Tilastollinen pakettipohjainen näytteenotto paketeille sekä aikapohjainen näytteenotto laskureille. [44, 58] Pakettivirtojen näytteenoton tulee tarjota jokaiselle havaitulle paketille yhdenvertainen mahdollisuus tulla näytteistetyksi, riippumatta pakettivirrasta, johon se kuuluu. Laskurinäytteiden ensisijainen tavoite on toimittaa verkkolaitteen laskurit määräajoin keräilijälle. sFlow-sanoman formaatti määrittelee standardoidun tavan näytteiden lähettämiseksi agentilta keräilijälle. sFlow-sanoman formaatti on määritetty käyttäen XDR (External Data Representation) standardia. XDR on kompaktimpi kuin ASN.1 ja tarjoaa yksinkertaisemman menetelmä koodata sekä purkaa sanomat. [58] Kuvassa 23 on esimerkki sFlow-protokollan virtausnäytteen tietorakenteesta.



Kuva 23. *Esimerkki sFlow virtausnäytteestä. Mukailtu lähteestä [58].*

5 VERKONVALVONNAN TOTEUTUS

Verkonvalvonta toteutettiin kuvan 24 mukaiseen verkkoympäristöön. Ympäristö koostuu yli 30 verkkokytkimestä. Kuvassa esitetyt verkkolaitteet sijaitsevat maantieteellisesti laajalle alueelle hajautettuna. Laitosten välinen etäisyys on pisimillään useita kymmeniä kilometrejä. Kuvan järjestelmän päälaitos on Laitos 1. Tässä laitoksessa sijaitsee automaatiojärjestelmään liittyvä päävalvomo, josta ohjataan myös muiden laitosten (2-10) prosesseja. Alaluvussa 5.1 toteutettu verkonvalvontasovellus sijaitsee päälaitoksessa. Laitosten välinen kommunikointi on toteutettu langattomien radiolinkkien avulla, jotka ovat asiakkaan omassa ylläpidossa.



Kuva 24. *Verkkoympäristö, johon verkonvalvontajärjestelmä toteutettiin.*

Esimerkki tällaisesta langattomasta radiolinkistä on Satel Oy:n valmistama Satellar XT 5RC IP-radioreititin. Satellar IP-radioreititin tarjoaa luotettavan tiedonsiirtoyhteyden toimintakriittisiin sovellusympäristöihin, jotka edellyttävät vakautta, korkeaa saatavuutta, turvallisuutta sekä pitkää kantamaa. IP-radioreititin toimii UHF (Ultra High Frequency) -radiotaajuudella ja mahdollistaa useiden kymmenien kilometrien kantaman. [59] IP-radioreitin tukee TCP/IP-protokollapinon mukaisia protokollia, kuten TCP, UDP, ICMP sekä SNMP -protokollia. Laitteessa on valittavana myös kaksi eri salausvaihtoehtoa, jotka ovat AES128 (Advanced Encryption Standard) ja AES256 -lohkosalausmenetelmät. [60]

Toteutetussa verkonvalvonnassa ei huomioitu laitteita, jotka sijaitsevat liitettyinä valvottaviin verkkokytkimiin. Palvelimien, logiikoiden sekä muiden automaatiolaitteiden valvonta on kuitenkin huomioitu verkonvalvontaprotokollia valitessa ja järjestelmän toteutusta suunniteltaessa. Automaatiolaitteiden rajoittuneet ominaisuudet verkonvalvontaprotokollien tukemisessa luovat kuitenkin haasteita. Esimerkiksi logiikoiden tarjoamat SNMP MIB -tiedostot ovat rajoittuneita, eikä niistä saada hallintatietoa yhtä tehokkaasti kuin perinteisistä IT-verkkolaitteista. Automaatiolaitteista saadaan rajallisesti kunnon-

valvonnan tarpeisiin soveltuvia objekteja. Käyttämällä IT-laitteita automaation asettamien mahdollisuuksien rajoissa, voidaan automaatioverkosta kerätä tietoa logiikoiden ja kenttälaitteiden välisestä tietoliikenteestä, esimerkiksi virtauksenseuranta teknologioiden avulla. Yksi lisämahdollisuus automaation Ethernet-väyläratkaisujen valvomiseksi voisi olla tämän diplomityön ulkopuolelle rajautuva pakettienkaappaus ja pakettien sisällön analysointi.

Verkonvalvonta toteutettiin keskitetyllä arkkitehtuurilla. Tämän arkkitehtuurin hyötyjä on helpompi hallittavuus sekä nopea käyttöönotto. Haittapuolena voidaan pitää valvontatiedon keruuta verkkoyhteyksien ollessa häiriintyneenä. Järjestelmän toteuttaminen hajautetun tai hierarkkisen verkonvalvonnan avulla on myös mahdollista. Arkkitehtuuria voidaan myös muokata lisääntyvän suorituskykytarpeen ilmetessä.

Valvontajärjestelmä toteutettiin interaktiivisena, joka tarkoittaa, että valvontaa suoritetaan jatkuvasti passiivisena ja aktiivisena valvontana. Järjestelmä ei kuitenkaan suorita automaattisia toimenpiteitä ongelmien korjaamiseksi tai niiden juurisyiden eristämiseksi. Nämä toimenpiteet tapahtuvat laitteiston ja järjestelmän ylläpitäjän toimesta. Alaluvussa 5.2 tarkastellaan järjestelmän testausta ja testauksen tuloksia. Alaluvussa 5.3 käsitellään tämän työn pohjalta löydetyt jatkotutkimus ja -kehitys kohteet.

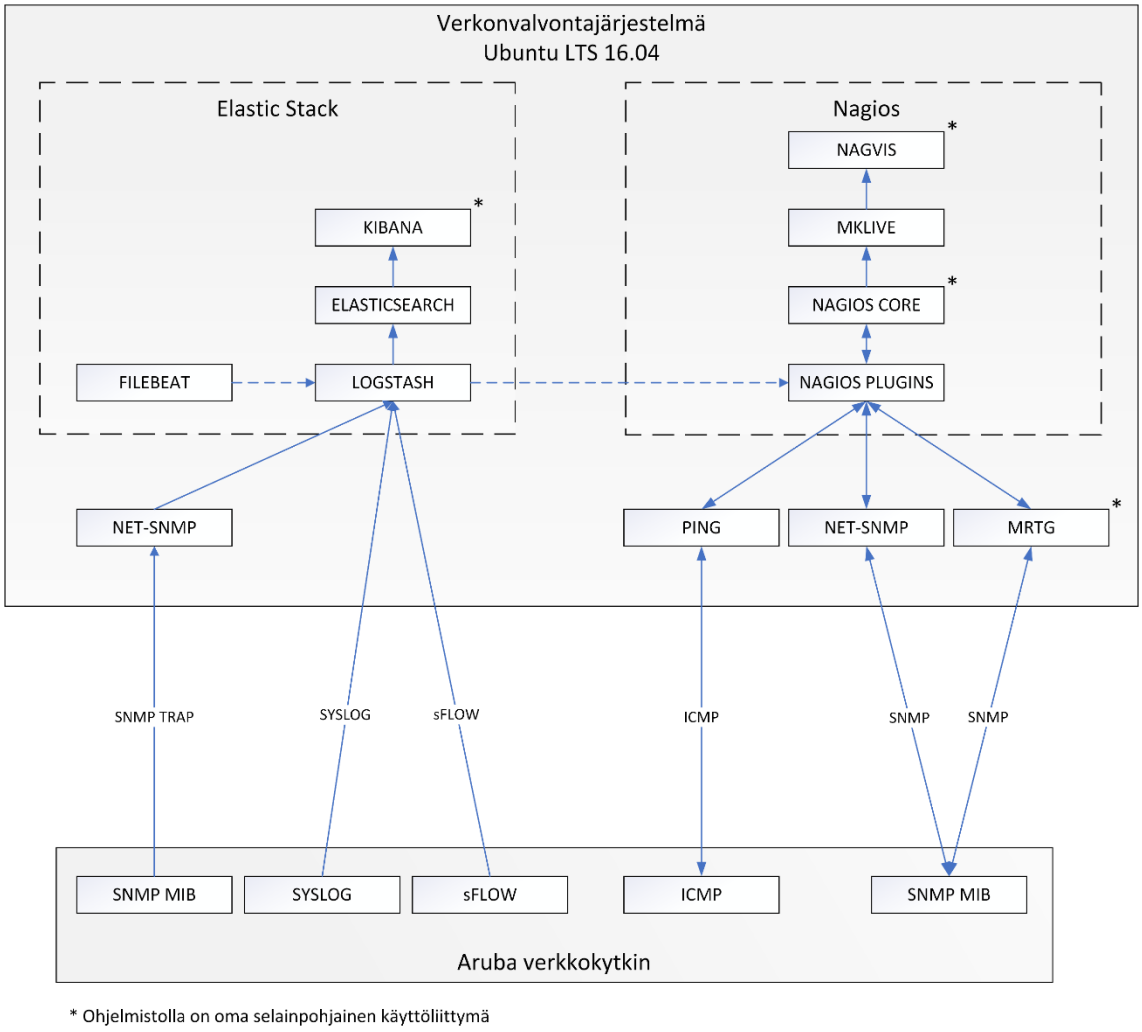
5.1 Verkonvalvontasovellus

Verkonvalvontasovellus toteutettiin Linux Ubuntu -käyttöjärjestelmälle. Toteutus ei kuitenkaan ole sidoksissa kyseiseen Linux versioon. Verkonvalvontakokonaisuutta lähdettiin suunnittelemaan kahdesta lähtökohdasta, jotka esitellään seuraavaksi.

Ensimmäinen lähtökohta oli kyselypohjaisen verkonvalvonnan toteuttaminen. Tähän tehtävään valittiin Nagios-sovellus. Nagioksen toimintaa täydentämään ja visuaalisuusvaatimusta toteuttamaan valittiin Nagioksen kanssa yhteistyöhön suunniteltu Nagvis-sovellus. Tämän lisäksi Nagios tarvitsee joukon liitännäisohjelmia, joiden avulla varsinaiset valvontatoiminnot voidaan suorittaa. Alaluvussa 5.1.1 käsitellään tarkemmin Nagios-sovellusta ja sen kanssa yhteistyössä toimivaa kokonaisuutta.

Toinen lähtökohta oli tapahtumapohjaisen verkonvalvonnan toteuttaminen. Tähän tehtävään valikoitui Elastic Stack -ohjelmistopino. Elastic Stack -ohjelmistopino ei ole varsinaisesti erikoistunut verkonvalvontatehtäviin, mutta sen avulla oli mahdollista toteuttaa tapahtumapohjainen lokien ja hälytysten keruujärjestelmä. Toinen valintaan vaikuttanut asia oli mahdollisuus kerätä ja analysoida virtausteknologioiden avulla kerättäviä suorituskyvyn valvontaan liittyviä tietoja. Myös tilaajayrityksen olemassa olevien Elastic Stack -ohjelmistopinolla toteutettujen ominaisuuksien implementoiminen tulevaisuudessa vaikutti valintaan. Alaluvussa 5.1.2 on esitetty tarkemmin Elastic Stack ja siihen sisältyvät ohjelmistot.

Verkonvalvontakokonaisuuteen on mahdollista valita yksi tai useampi edellä mainituista ominaisuuksista. Valvontasovellus voidaan ottaa käyttöön niin, että se sisältää joko kyselypohjaisen verkonvalvontamenetelmän, tapahtumapohjaisen lokien ja hälytysten keruujärjestelmän tai suorituskyvyn valvonnan virtausteknologioiden avulla. Näistä kolmesta ominaisuudesta on myös mahdollista koostaa haluttu kokonaisuus tai ottaa käyttöön ne kaikki. Kuvassa 25 on esitetty komponentit, joista verkonvalvonnan sovelluskokonaisuus koostuu. Kuvassa ilmenee myös sovellusten välinen ja järjestelmän sisäinen kommunikointi sekä Aruba merkkisen verkkokytkimen ja verkonvalvontasovellusten välinen kommunikaatio.

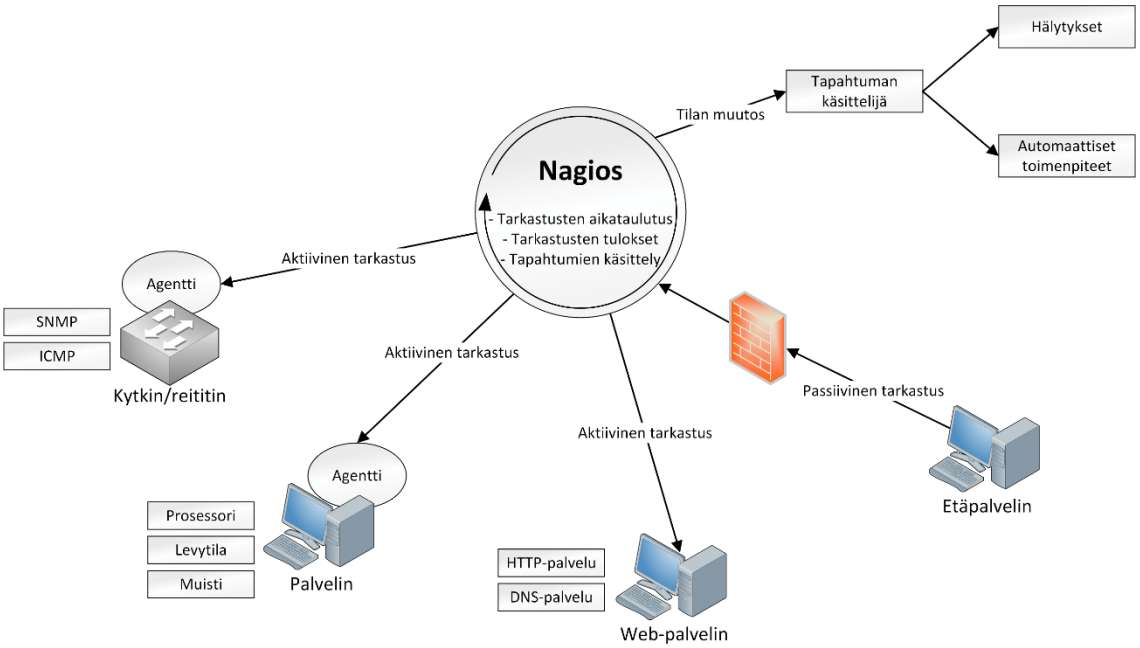


Kuva 25. *Työssä koostetun verkonvalvontakokonaisuuden arkkitehtuuri.*

5.1.1 Nagios

Verkonvalvontasovelluksen kyselypohjainen valvonta toteutettiin Nagios Core -sovelluksella. Nagios Core -sovellus on ilmainen avoimen lähdekoodin työkalu verkon- ja järjestelmänvalvontaan. Sovelluksen pohjalta on myös kehitetty kaupallinen versio nimeltä Nagios XI. Kaupalliseen versioon on kerätty käyttäjälle valmiiksi liitännäisso-

velluksia sekä muita ominaisuuksia, joiden ansiosta sen käyttöönottaminen on hieman yksinkertaisempaa. Käytön aloittaminen ja järjestelmän konfigurointi Nagios Core -versiossa on verrattain monimutkaisempaa, mutta ominaisuuksia ja liitännäissovelluksia löytyy laajan kehitysyhteisön ansiosta runsaasti. Seuraavaksi käydään läpi kyselypohjaisen valvonnan toteutuksessa käytetyt sovelluskomponentit sekä selvitetään, mitä komponenttien avulla tehtiin. Kuvassa 26 on esitetty Nagios-prosessi sekä siihen liittyvät aktiiviset ja passiiviset tarkastukset.



Kuva 26. *Nagios-prosessi. Mukailtu lähteestä [61].*

Nagios Core

Nagios Coren (myöhemmin Nagios) avulla voidaan valvoa verkkolaitteiden toimintaa. Nagios suorittaa jatkuvasti sille määritellyt testit ja tarkastaa testien tuloksien avulla verkon laitteiden tilan sekä toiminnan. Nagioksen valvonnan tarkoituksena on tunnistaa mahdollisimman nopeasti laitteet, jotka eivät toimi odotetulla tavalla. [62] Näitä Nagioksen suorittamia testejä kutsutaan aktiivisiksi testeiksi. Nagios voi toimia myös kuuntelijana testituloksille, jolloin testit suoritetaan muualla ja valmiit tulokset lähetetään Nagiokselle. Tätä kutsutaan passiiviseksi testaamiseksi. [61]

Nagios jakaa valvonnan kahteen kategoriaan, joita ovat isäntien- ja palveluidenvalvonta [61, 62]. Isännällä tarkoitetaan verkossa olevia fyysisiä tai virtuaalisia laitteita, kuten kytkimiä, reitittimiä ja palvelimia. Palvelulla tarkoitetaan esimerkiksi palvelimella suoritettavaa SSH (Secure Shell) prosessia. [62] Nagios reagoi isäntien ja palveluiden tilan muutokseen tai tapahtumaan käyttämällä tapahtumakäsittelijöitä. Tapahtumakäsittelijöiden avulla voidaan lähettää esimerkiksi sähköposti tai suorittaa skripti, joka pyrkii korjaamaan ongelman automaattisesti. [61]

Nagios suorittaa kaikki valvontatehtävät erillisillä lisäohjelmilla. Lisäohjelmat ovat käytännössä skriptejä sekä pieniä sovelluksia, joita Nagios suorittaa määritetyin aikavälein. Liitännäissovellukset ovat vastuussa varsinaisten valvontatoimintojen suorittamisesta ja tulosten analysoimisesta. Valmiita liitännäissovelluksia on tarjolla lukuisia, mutta niitä voi myös rakentaa itse. Tämä laajentaa Nagioksen käyttömahdollisuuksia ja sen ansiosta voidaan valvoa lähes mitä tahansa. [61, 62] Lisäohjelmat voidaan kirjoittaa millä tahansa ohjelmointikielellä [62].

Nagios valvontajärjestelmä voidaan toteuttaa alaluvussa 4.2 esitettyjen arkkitehtuurien avulla. Se voidaan toteuttaa keskitetyn arkkitehtuurin lisäksi myös hajautettuna tai hierarkkisena. Valvonta voidaan toteuttaa useiden Nagios-palvelimien avulla, jotka valvovat omia verkkosegmenttejä. Hajautetut Nagios-palvelimet lähettävät tulokset keskitetylle Nagios palvelimelle, joka kerää kootusti kaikkien suoritettujen testien tulokset. [62]

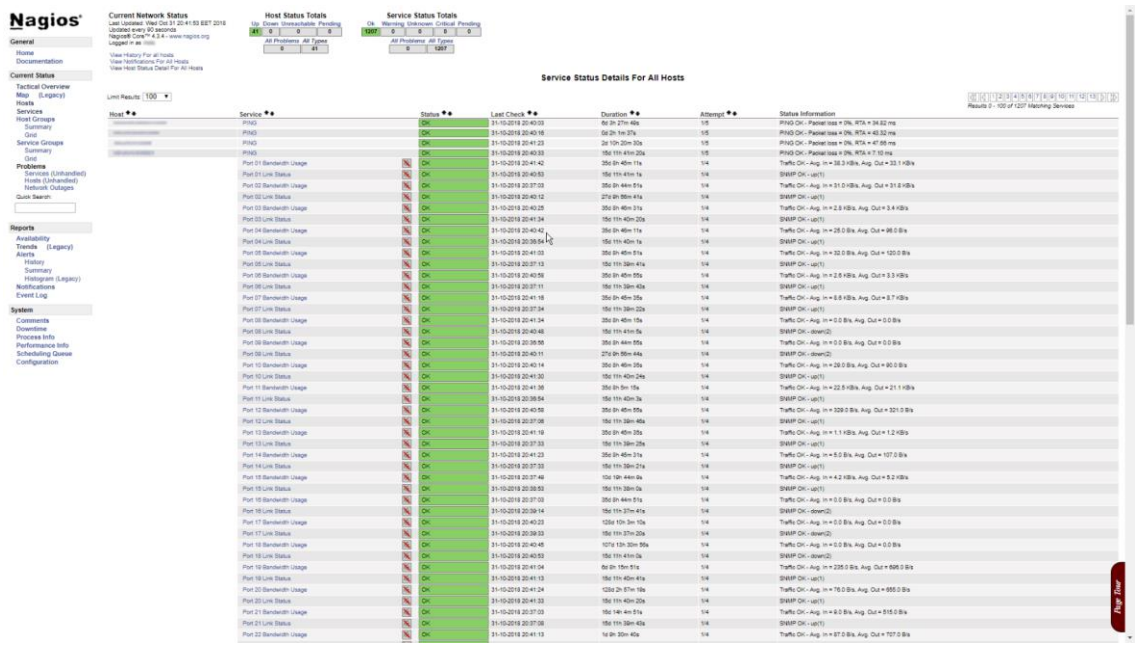
Nagioksen selainpohjainen käyttöliittymä tarjoaa sisäänrakennetun raportoinnin. Raportointi sisältää esimerkiksi saatavuusraportin, trendit laitteiden tiloista sekä hälytystiedot. Usein Nagiosta laajennetaan myös tässä yhteydessä lisäosilla, jotka parantavat historiatrendien seuraamista ja visualisoi tiedot tehokkaammin. Lisäosat hyödyntävät Nagioksen keräämiä tietoja ja esittävät ne erilaisilla visualisointityökaluilla. Useimmiten tiedot tallennetaan tässä yhteydessä RRD (Round Robin Database) -tiedostoon. Raportoinnin parantamiseen soveltuvia lisäosia ovat esimerkiksi PNP4Nagios sekä Nagiosgraph. [61]

Nagiosta käytettiin tässä tutkimuksessa aktiivisten verkonvalvontaoperaatioiden suorittamiseen TMN-viitekehyksen elementinhallinnan tasolla. Nagioksen avulla toteutettiin FCAPS-toimintamallin mukainen vikojen valvonta huomioiden verkonvalvonnalle asetetut vaatimukset. Kyselypohjaista verkonvalvontaa lähdettiin toteuttamaan kahdella tarkoitukseen soveltuvalla verkonvalvontaprotokollalla. Valitut protokollat olivat SNMP sekä ICMP.

Nagios konfiguroitiin suorittamaan kyselyt SNMP-protokollan avulla. Kyselyt suoritetaan verkkolaitteille sopivin määräajoin, valvottavasta objekteista riippuen kyselyitä suoritetaan 1-5 minuutin välein. Verkkolaitteilta kerätään hallittavien objektien arvot, kuten rajapinnan porttien tilat ja pakettivirheet, prosessorin kuormitus, laitteen lämpötila sekä siirtoyhteyskerrokselle liittyvien laskureiden arvot, kuten CRC (Cyclic Redundancy Check), FCS (Frame Check Sequence) ja Duplex mismatch.

MRTG (Multi Router Traffic Grapher**)** -ohjelmiston avulla muodostetaan verkkolaitteen rajapinnan liikennemäärien lokitiedostot. MRTG-ohjelmisto ja sen toiminta esitellään myöhemmin tässä luvussa. Nagioksen avulla valvotaan liikennemääriä seuraamalla MRTG:n tuottamien lokitiedostojen sisältöä.

Nagioksen avulla suoritetaan jokaiselle verkkolaitteelle ICMP echo -kyselyt sopivin väliajoin, tässä tapauksessa kahden minuutin välein. Tällä tavalla varmistetaan laitteen tavoitettavuus verkossa sekä pystytään samalla valvomaan viivettä ja pakettien katoamista verkossa. Kaikille Nagioksen avulla valvottaville asioille määritellään raja-arvot, joiden ylittyessä muodostetaan varoituksia sekä hälytyksiä. Nagioksen hälytykset toteutettiin web-käyttöliittymän tarjoamien hälytyksien lisäksi Nagvis-visualisointilisäosalla sekä sähköpostihälytyksinä. Kuvassa 27 on kuvakaappaus Nagioksen käyttöliittymästä. Kuvassa näkyy neljän verkkolaitteen ICMP echo -kyselyiden tila ja tulokset, sekä yhden verkkolaitteen rajapinnan porttien tilat ja liikennemäärät.



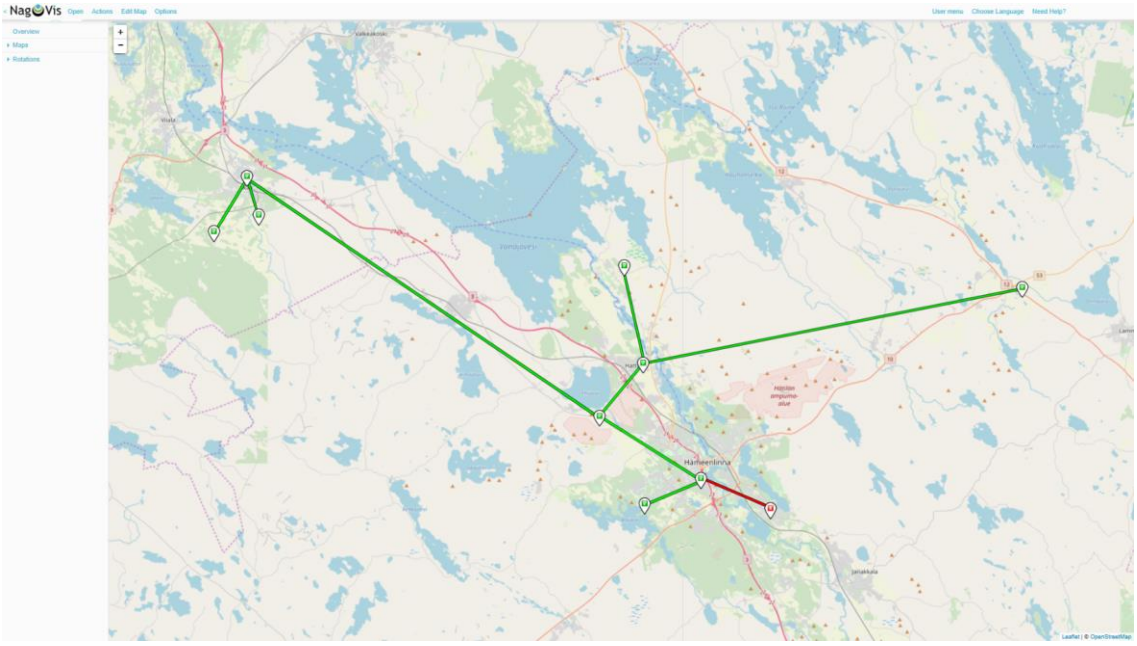
Kuva 27. *Nagioksen selainpohjainen käyttöliittymä, jossa näkyy valvonnan kohteena olevia laitteita sekä palveluita.*

NagVis

Nagvis on Nagioksen valvontatietojen visualisointiin tarkoitettu lisäosa. Nagvis on avoimen lähdekoodin toteutus. Nagvisin avulla voidaan esittää Nagioksen valvontatiedot interaktiivisten karttojen ja ikonien avulla. Nagvis tarjoaa selainpohjaisen käyttöliittymän, jonka avulla karttoja voidaan luoda sekä muokata. [63] Karttoja voidaan luoda ja muokata selaimen lisäksi myös tekstipohjaisina määrittelytiedostoina.

Nagvis tarjoaa Nagiokseen määritetyt verkkolaitteet ja palvelut automaattisesti, eikä niitä tarvitse määritellä uudelleen Nagvisia käyttöönotettaessa. Karttapohjana voidaan käyttää esimerkiksi laitoksen pohjakuvaa tai maantieteellistä karttaa. Maantieteellisen kartan avulla voidaan helposti havainnollistaa maantieteellisesti hajautettujen verkkolaitteiden tilaa yhdellä näkymällä. Kuvassa 28 on kuvakaappaus Nagvisin maantieteellisestä karttanäkymästä. Kuvassa näkyy yleisnäkymä laitosten verkkolaitteiden tiloista. Nagvisin kartat voivat olla hierarkkisia, jolloin yleisnäkymästä päästään siirtymään aina

yksityiskohtaisempiin kuviin. Nagvisin ja Nagioksen yhteistyö toimii saumattomasti ja Nagvisin näkymästä päästään siirtymään Nagioksen käyttöliittymään, painamalla kartalta verkkolaitetta tai palvelua. Nagvis tarjoaa mahdollisuuden omien ikonien käyttämiseen sekä karttanäkymien ja tilatietojen kooste pop-up -ikkunoiden muokkaamiseen.



Kuva 28. *Nagvisin selainpohjainen käyttöliittymä, jossa näkyy laitosten maantieteellinen sijainti, laitoskohtaisten verkkolaitteiden yhdistetyt tilatiedot sekä laitosten välisien yhteyksien tilat.*

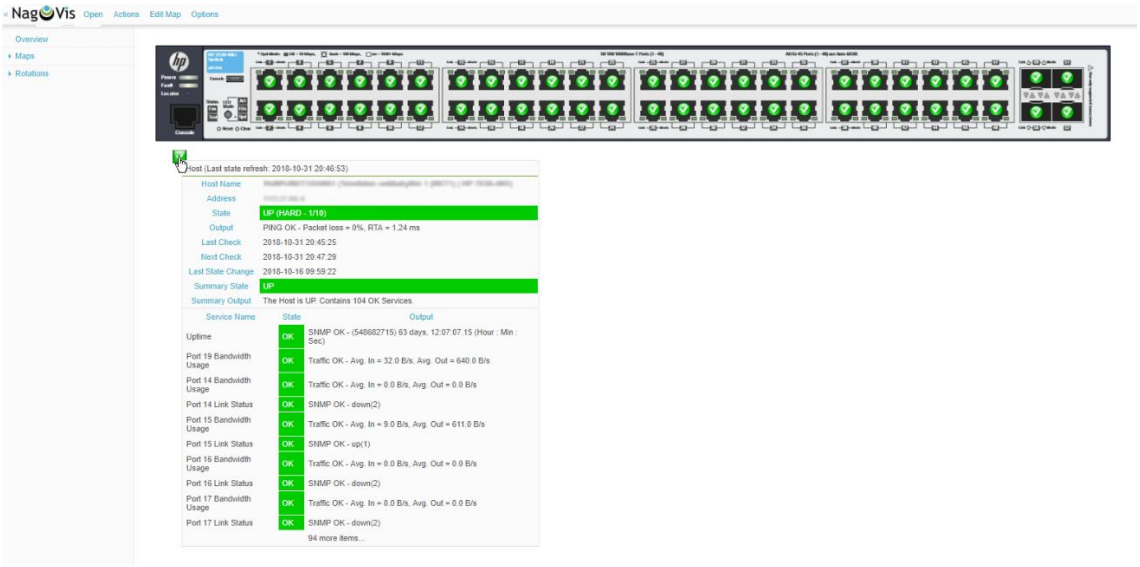
Nagvis tarvitsee taustalle ohjelmiston, jonka avulla Nagioksen valvontatiedot saadaan käytettäviksi ulkopuolisessa ohjelmassa [63]. MK Livestatus käyttää Nagios Event Broker API (Application programming interface) -rajapintaa, jonka avulla tiedot luetaan suoraan Nagioksen sisäisistä tietorakenteista. [63, 64] MK Livestatus on erittäin yksinkertainen moduuli Nagioksen tapahtumien välittämiseksi. MK Livestatus sisältyy uusimpiin Nagvis versioihin. Nagvis taustaohjelmia on olemassa muutama erilainen vaihtoehto, mutta ne eivät suoriudu tehtävästä yhtä tehokkaasti ja tarvitsevat lisäksi erillisen tietokannan. [63]

Verkon visualisointi ja ongelmakohtien esittäminen selkeästi graafisen käyttöliittymän avulla verkon ylläpitäjälle, oli vaatimuksena toteutettavalle valvontajärjestelmälle. Nagvis tarjoaa työkalut tämän vaatimuksen toteuttamiseksi. Nagvisin avulla pystytään tarjoamaan erilaisia graafisia käyttöliittymiä ja näyttämään verkon osa-alueiden kootut sekä yksityiskohtaiset tilatiedot.

Nagvisin avulla luotiin erilaisia karttoja, jotka kuvaavat verkon fyysisiä topologioita, verkkolaitteita sekä verkkolaitteiden sijaintia kyseisessä järjestelmässä. Nagvisin pääkuvaksi toteutettiin maantieteellinen karttanäkymä, josta käy ilmi laitosten sijainnit ja

laitosten välisten yhteyksien tilat. Kartan avulla esitetään myös jokaisen laitoksen verkkolaitteiden tilojen yhteenveto.

Jokaisesta laitoksesta toteutettiin laitoskohtaisten laitteiden koostesivu, jossa näkyy kyseisen laitoksen kaikkien verkkolaitteiden tilat. Alimmaksi kartaksi hierarkiassa toteutettiin yksittäisten verkkolaitteiden tiloja kuvaavat sivut, esimerkki tällaisesta sivusta on esitetty kuvassa 29. Tältä alimmalta kartalta voidaan siirtyä Nagioksen käyttöliittymään, josta nähdään vielä tarkemmat tiedot valvottavasta verkkolaitteesta tai sen palvelusta. Nagvisin avulla toteutettiin käytännönläheinen ja selkeä kuvaus verkosta, verkon laitteista sekä laitteiden välisistä suhteista.



Kuva 29. *Nagvisin selainpohjainen käyttöliittymä, jossa näkyy yhden verkkolaitteen yleinen tila sekä rajapintakohtaisten palveluiden tilat.*

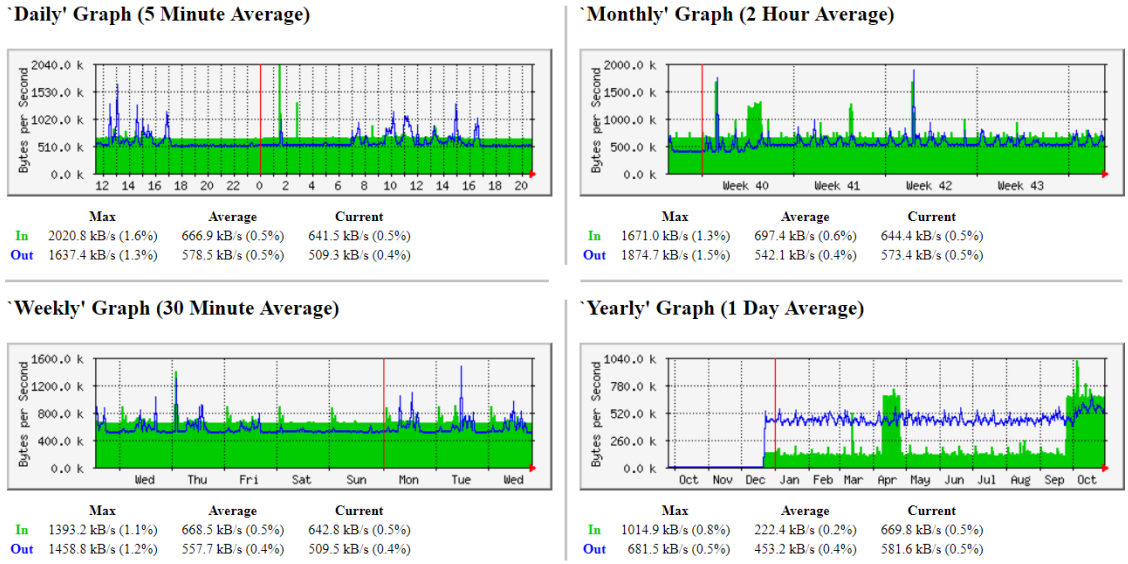
Multi Router Traffic Grapher (MRTG)

MRTG on avoimen lähdekoodin sovellus, jota on yleisesti käytetty verkkolaitteiden liikennemäärien keräämiseksi. MRTG kerää verkkolaitteiden liikennemäärät SNMP-protokollan avulla. MRTG ylläpitää lokitiedostoja laitteilta kerätyistä arvoista. Näitä lokitiedostoja voidaan käyttää hyväksi verkonvalvonnassa. Nagioksen avulla voidaan seurata ja valvoa MRTG:n lokitiedostojen arvoja. Nagios vertailee MRTG:n lokitiedostossa olevia arvoja asetettuihin raja-arvoihin. Raja-arvojen ylityksestä voidaan suorittaa Nagiokselle tyypilliset toimenpiteet, kuten automaattinen tehtävän käynnistys tai hälytyksen muodostus. [65]

MRTG:n avulla on mahdollista luoda automaattisesti päivittyviä graafisia kuvaajia. Tarjolla on neljä kuvaajaa, jotka piirtävät lokitiedoston arvot päivä, viikko, kuukausi sekä vuosi tasolla. MRTG tuottaa myös selainpohjaisen käyttöliittymäsivun, jonka avulla graafisia kuvaajia voidaan seurata. MRTG:n käyttö ei kuitenkaan rajoitu ainoastaan verkkoliikenteen keräämiseen. MRTG:n avulla on mahdollista kerätä vapaasti valittujen

SNMP-objektien arvoja. MRTG-sovellusta on mahdollista käyttää yhteistyössä RRDtool-sovelluksen kanssa. RRDtool-sovellus esitellään jäljempänä tässä luvussa. Näiden sovellusten yhteistyössä MRTG ei tuota grafiikoita, vaan kerää ainoastaan tiedot verkkolaitteilta ja tallentaa ne RRD-tietokantaan. RRDtool-sovelluksen avulla huolehditaan grafiikoiden tuottamisesta sekä tietokannan ylläpidosta. [65]

MRTG-sovellusta käytettiin verkkolaitteiden rajapinnan liikennemäärien laskureiden kyselyyn SNMP-protokollan avulla. MRTG:n avulla lasketaan verkkolaitteiden liikennemäärien kumulatiivisista laskureista tulevan sekä lähtevän liikenteen nykyinen arvo. Liikennemäärät luetaan verkkolaitteilta viiden minuutin välein. Jokaisen luennan yhteydessä arvot päivitetään MRTG:n lokitiedostoihin sekä luodaan päivitetyt graafiset kuvaajat. Vaatimuksena oli seurata liikennemääriä. Nagioksen avulla valvotaan lokitiedostojen arvoja ja seurataan että ne eivät ylitä asetettuja raja-arvoja. MRTG:n luomat kuvaajat tuottavat lisäarvoa ja niiden avulla voidaan tarkastella pitkällä aikavälillä liikennemäärien kehittymistä. Kuvassa 30 on esitetty MRTG:n luomat graafiset kuvaajat verkkolaitteen yhdestä rajapinnasta.



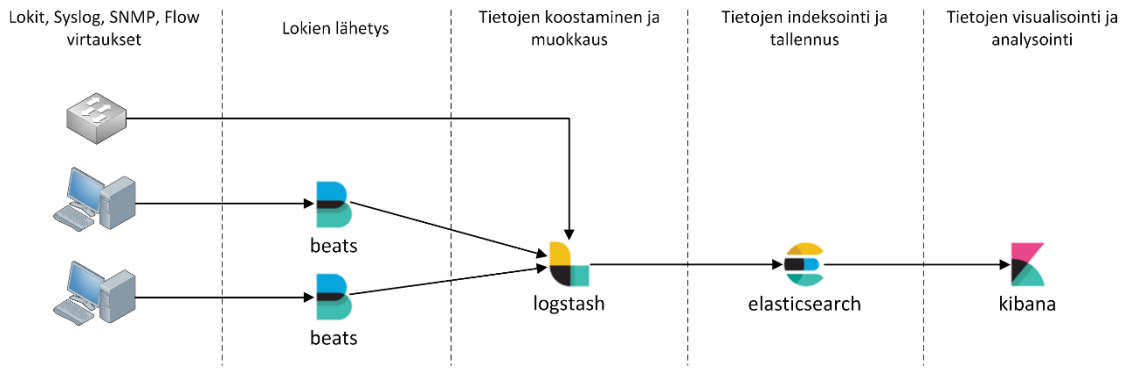
Kuva 30. *MRTG:n luomat kuvaajat verkkolaitteen rajapinnan liikennemääristä.*

5.1.2 Elastic Stack

Elastic Stack -sovelluskokonaisuus koostuu neljästä avoimen lähdekoodin sovelluksesta. Nämä sovellukset ovat Elasticsearch, Logstash, Beats ja Kibana. Käytännössä nimi Beats viittaa useisiin toimintaperiaatteeltaan samankaltaisiin sovelluksiin. [66] Virallisesti tuettuna on kuusi Beats-ohjelmaa, jotka ovat Auditbeat, Filebeat, Heartbeat, Metricbeat, Packetbeat sekä Winlogbeat [67]. Elastic Stack -sovelluksia kehittää, hallinnoi sekä ylläpitää Elastic-ohjelmistoyritys [68].

Elasticsearch on hakumoottori sekä NoSQL-tietovarasto, joka pohjautuu Apache Lucence hakumoottoriin. Logstash on tiedonkeruutyökalu, joka kykenee vastaanottamaan eri lähteistä tulevaa tietoa, suorittamaan tiedon muotoilun sekä lähettämään käsitellyn tiedon tallennettavaksi tai analysoitavaksi. Kibana on tiedon visualisointiin erikoistunut ohjelmisto, joka toimii Elasticsearch-sovelluksen päällä. [68] Beats-ohjelmat ovat operatiivisen tiedon lähettäjiä. Beats-ohjelmat lähettävät tiedot valvottavalta palvelimelta suoraan tallennettavaksi Elasticsearchille tai muokattavaksi Logstashille. [67]

Elastic Stack -sovelluskokonaisuutta käytetään yleisesti IT-ympäristöissä keskitettynä lokien keruu järjestelmänä. Kuvassa 31 on esitetty Elastic Stack -ohjelmistokokonaisuus, jota käytetään lokitietojen keräämiseen. Beats-sovellusten avulla lähetetään seurattavien verkkolaitteiden lokit Logstashille. Logstashin avulla muokataan vastaanotettu lokisanoma haluttuun muotoon. Tämän jälkeen lokitiedot indeksoidaan ja tallennetaan Elasticsearchin toimesta. Lopuksi Kibanan avulla voidaan etsiä, esittää sekä analysoida tietoja visuaalisessa muodossa. Elastic Stack -sovelluskokonaisuus on noussut suureen suosioon, koska se vastaa lokien analysoinnin tarpeisiin ja tuo vaihtoehdon kaupallisille sovelluksille. [68]



Kuva 31. *Elastic Stack -ohjelmistokokonaisuus lokitietojen keräämiseksi. Mukailtu lähteestä [68].*

Elastic Stack -sovelluskokonaisuutta käytettiin tässä tutkimuksessa keskitetyn lokien keruu järjestelmän luomiseen. Lokien keruu järjestelmä toimii TMN-viitekehyksen elementinhallinnan tasolla. Lokien keruu järjestelmän avulla toteutettiin FCAPS-toimintamallin vikojen valvonnan osa-alue, joka vastaa tapahtumapohjaiseen vikojen valvontaan. Elastic Stack -sovelluskokonaisuuden avulla kerättiin myös kokeilumielessä virtausteknologioiden avulla tuotettuja virtaustallenteita. Virtaustallenteiden avulla voidaan tuottaa arvokasta tietoa TMN-viitekehyksen elementinhallinnan sekä verkonhallinnan tasoille. Virtaustallenteiden avulla pystytään vastaamaan FCAPS-toimintamallin suorituskyvyn valvonnan sekä vikojen valvonnan osa-alueisiin.

Filebeat

Filebeat on kevyt ohjelmisto lokitietojen välittämiseen. Filebeat seuraa haluttuja lokitiedostoja, kerää niistä tapahtumat ja lähettää ne eteenpäin. Filebeat koostuu kahdesta pää-

komponentista, jotka ovat syötteet (inputs) sekä keräilijät (harvesters). Nämä komponentit toimivat yhteistyössä. Syöte-komponentti on vastuussa keräilijöiden hallinnasta ja tietolähteiden etsimisestä. Syöte-komponentti tarkastaa kaikki tiedostot ja päättää tarvitseeko keräilijä käynnistää, onko keräilijä jo käynnissä vai voiko tiedoston ohittaa. Yksi keräilijä on aina vastuussa yhden tiedoston sisällön lukemisesta. [69]

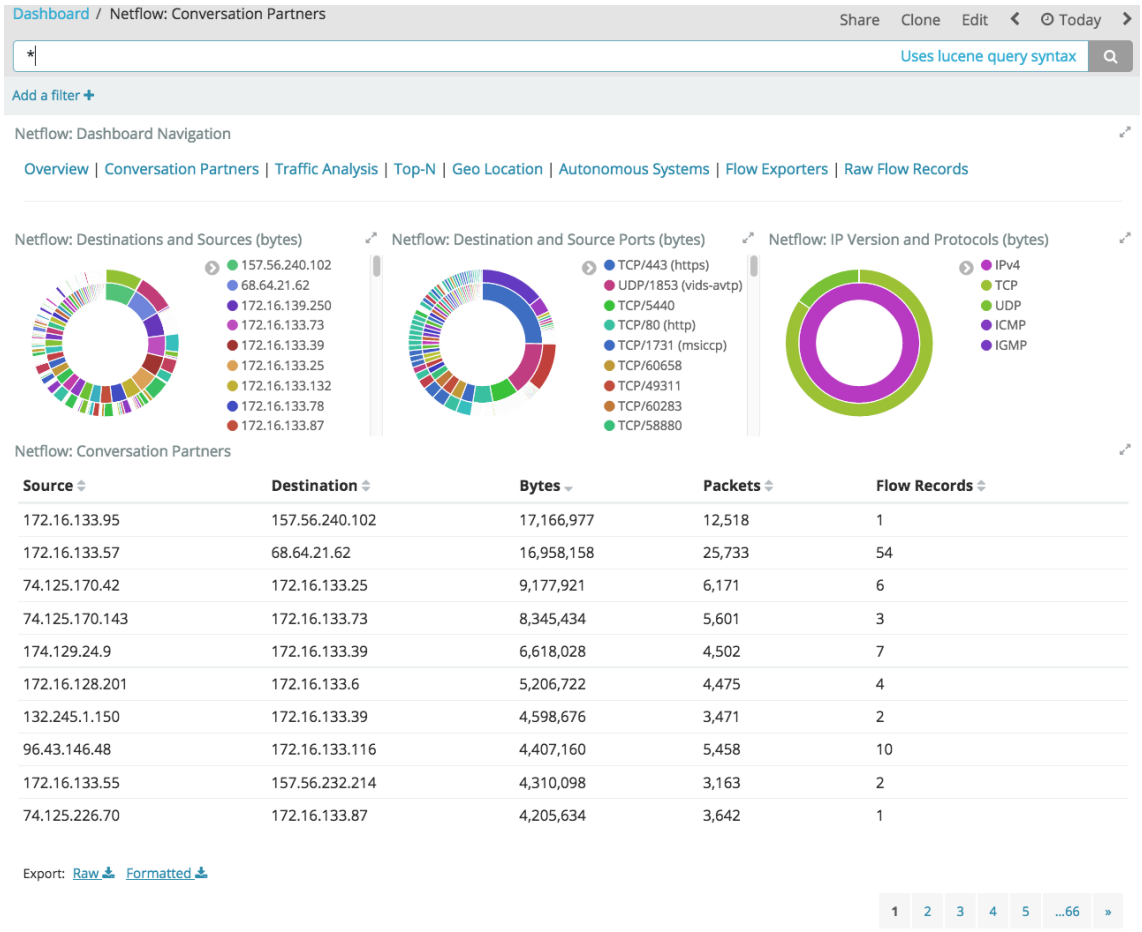
Keräilijät lukevat tiedostot rivi kerrallaan ja lähettävät tiedot eteenpäin. Filebeat ylläpitää rekisteriä tiedoston tilasta, eli siitä mitkä tiedoston rivit on viimeksi luettu. Keräilijä jatkaa aina siitä mihin on viimeksi jäänyt. Samalla varmistetaan, että kaikki tiedoston rivit tulee lähetettyä. Jos lähetyksiä vastaanottava osapuoli ei ole tavoitettavissa Filebeat pitää kirjaa siitä, mitkä rivit ovat viimeksi lähetetty. Tietojen lähettämistä jatketaan välittömästi, kun vastaanottaja on jälleen tavoitettavissa. [69]

Filebeat-sovellusta ei käytetty tämän diplomityön puitteissa, koska työtä ei laajennettu koskemaan palvelinten valvontaa. Filebeat-sovellus on kuitenkin huomioitu ja se näkyy osana verkonvalvontajärjestelmän sovelluskokonaisuutta kuvassa 25. Filebeat tulee osaksi järjestelmää välittömästi, kun palvelimien lokitietojen kerääminen halutaan aloittaa.

Logstash

Logstash on tiedonkeräysmoottori, joka mahdollistaa tietojen keräämisen ja muokkaamisen lähes reaaliaikaisesti [70]. Logstash perustuu liukuhihna-arkkitehtuuriin (pipeline architecture). Logstashin avulla voidaan kerätä tietoa useilta lähteiltä samanaikaisesti, muokata tietoja sekä lähettää muokatut tiedot tallennettavaksi, esimerkiksi Elasticsearchille. [71] Logstash on alun perin kehitetty lokitietojen keräämiseen, mutta sen ominaisuuksia on laajennettu koskemaan kaikenlaisten tietovirtojen keräämistä.

Logstashin avulla voidaan dynaamisesti yhdistää tiedot erilaisista lähteistä sekä normalisoida ne haluttuun muotoon [70]. Tämän jälkeen tietoa voidaan etsiä sekä analysoida tehokkaammin liiketoiminnan arvon tuottamista varten. Logstashin arkkitehtuuri mahdollistaa yli 200 valmiin moduulin käytön, joista yksi esimerkki on NetFlown keräämiseen tarkoitettu moduuli. Kuvassa 32 on esimerkki NetFlow-moduulin avulla kerätyistä virtaustallenteista visualisoituna Kibanassa. [71]



Kuva 32. *Logstash-moduuli NetFlow-virtaustallenteiden keräämiseksi ja visualisoimiseksi Kibanassa. [70].*

Tässä tutkimuksessa kerättiin Logstashin avulla verkkokytkimien Syslog ja SNMP Trap -sanomia. Syslog-sanomat kerättiin verkkokortin UDP-liikenteestä portista 514. Kerätyt sanomat muotoiltiin sopivaan muotoon ja niihin lisättiin ilmoituksen vakavuutta kuvaavan numeron lisäksi myös tekstimuotoinen vakavuuden kuvaus. SNMP Trap -sanomia oli tarkoitus lukea verkkokortin liikenteestä Logstashin tarjoamalla SNMP Trap -moduulilla. Moduulin avulla ei kuitenkaan saatu aikaiseksi toivottua tulosta. Moduuli ei kääntänyt OID-yksilöintitunnuksia eikä SNMP Trap -sanomia MIB-tiedostojen avulla selkokieliseen muotoon. Tästä johtuen SNMP Trap -sanomia päädyttiin keräämään NET-SNMP -sovelluksella. NET-SNMP -sovellus määritettiin kuuntelemaan SNMP Trap -sanomia verkkokortin UDP-liikenteestä portista 162. NET-SNMP suoriutuu moitteettomasti myös sanomien kääntämisestä selkokieliseen muotoon. Tämän jälkeen sanomat muokattiin vielä haluttuun muotoon Logstashin avulla. Sanomien muokkauksen jälkeen Logstash lähettää ne JSON (JavaScript Object Notation) -dokumentteina tallennettavaksi Elasticsearchiin.

Logstashin avulla kokeiltiin myös kerätä NetFlow ja sFlow -sanomia. Netflow sanomien kerääminen onnistuu kohtuullisen helposti käyttämällä valmista NetFlow-moduulia. sFlow-sanomien kerääminen onnistuu myös esimerkiksi Logstashin tarjoa-

man sFlow-koodekin avulla. Vastaavia suodattimia ja tiedon muotoilua on mahdollista rakentaa myös itse, esimerkiksi NetFlow, sFlow sekä IPFIX -sanomille. Tämä edellyttää kuitenkin syvällistä perehtymistä kyseisiin teknologioihin.

Elasticsearch

Elasticsearch on skaalautuva tekstihaku- ja analysointimoottori [72], joka toimii myös NoSQL-dokumenttien tietovarastona [66]. Sen avulla voidaan tallentaa, hakea sekä analysoida suuria määriä tietoa nopeasti ja lähes reaaliaikaisesti. Elasticsearchilla on useita käyttökohteita, esimerkiksi loki- ja tapahtumatietojen varastointi sekä analysointi. [72]

Elasticsearchin lähes reaaliaikaisuudella tarkoitetaan, että keskimääräinen viive dokumentin saapumisesta siihen, että se on käytettävissä, on noin yhden sekunnin mittainen. Elasticsearch voidaan suorittaa yhdellä palvelimella tai laajentaa sen klusteria useille palvelimille. Järjestelmä, joka koostuu useista palvelimista, mahdollistaa kuormituksen tasaamisen ja silti yhtenäisesti toimivan tietojen sälyttämisen, indeksoinnin sekä hakutoiminnon. Indeksoinnilla tarkoitetaan samanlaisia ominaisuuksia sisältävien dokumenttien kokoelmaa.

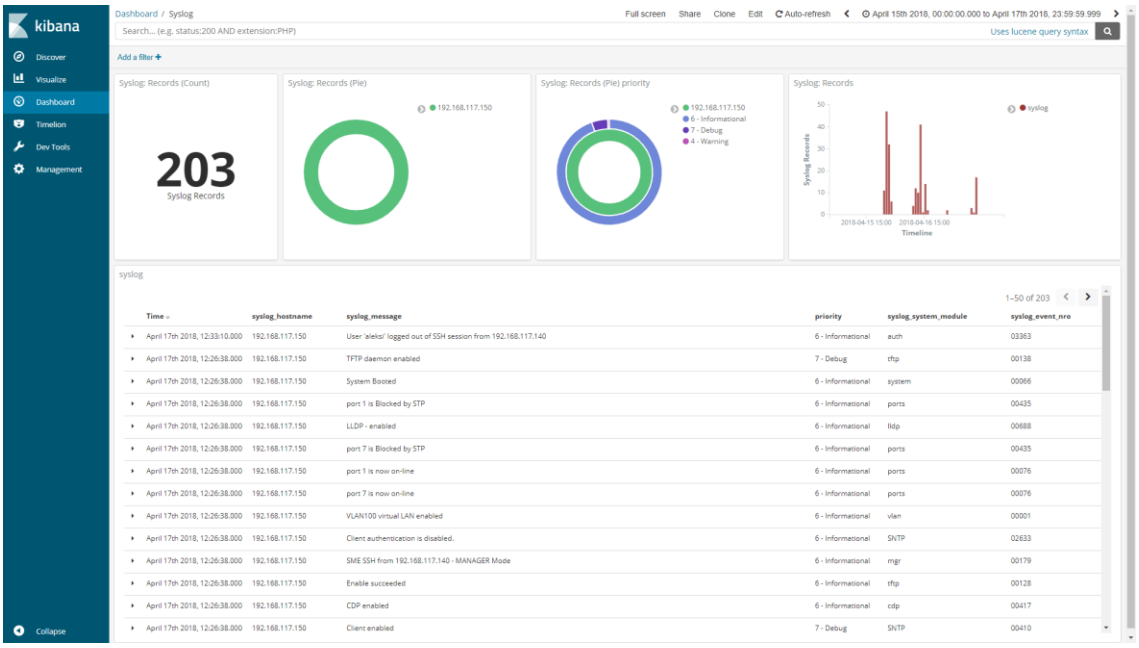
Tässä tutkimuksessa Elasticsearchia käytettiin tietovarastona, johon tallennetaan Syslog ja SNMP Trap -sanomat. Elasticsearchia käytettiin tallentamaan tietoa, jota halutaan myöhemmin etsiä ja esittää Kibanan analysointi- ja visualisointityökalun avulla. Virtausteknologioita testattaessa virtaustallenteet varastoitiin myös Elastcisearchin avulla. Elasticsearch tietovarasto vastaa valvontasovelluksen vaatimukseen, jonka mukaan alustan tulee olla skaalattavissa tulevaisuuden tarpeita varten. Liitteen B kuvassa 1 on esitetty Elasticsearchiin tallennettu JSON-dokumentti, joka sisältää sFlow-virtaustallenteen.

Kibana

Kibana on selainpohjainen analysointi- ja visualisointialusta, joka on suunniteltu käytettäväksi yhdessä Elasticsearchin kanssa. Kibanan avulla voidaan etsiä, tarkastella ja olla vuorovaikutuksessa Elasticsearchin indeksoitujen dokumenttien kanssa. Sen avulla voidaan suorittaa tiedon analysointia ja visualisoida tiedot erilaisten kaavioiden, taulukoiden sekä karttojen avulla. Sen selainpohjaisen käyttöliittymän avulla voidaan tehdä ja jakaa dynaamisia hallintapaneeleita (dashboards), jotka näyttävät ja päivittävät lähes reaaliaikaisesti Elasticsearch kyselyiden tulokset. [66]

Tässä tutkimuksessa Kibanaa käytettiin lokitietojen visuaaliseen esittämiseen. Kibanan avulla toteutettiin hallintapaneelit Syslog sekä SNMP Trap -sanomille. Hallintapaneelien avulla tuotetaan verkonvalvonnan tarpeisiin informaatiota havainnollisessa ja selkeässä muodossa. Kibanan avulla on mahdollista suodattaa ja tarkastella tietoja vapaasti valittavien suodatusperusteiden avulla. Lokitietoja voidaan tarkastella esimerkiksi erilaisilla aikaväleillä sekä suodattaa näkyviin vain tietyn tyyppiset tai tietyn laitteen lokisanomat. Suodatusominaisuudet auttavat verkon ongelmatilainteiden selvittämisessä,

jolloin voidaan tarkastella kaikkia ongelmatilanteen aikana tai sitä edeltäneellä ajanhetkellä vastaanotettuja lokisanomia. Näiden lokisanomien perusteella voidaan selvittää esimerkiksi ongelman juurisyitä. Kuvassa 33 on esitetty Kibanalla toteutettu Syslog-sanomien hallintapaneeli. Vastaava hallintapaneeli toteutettiin myös SNMP Trap -sanomille.

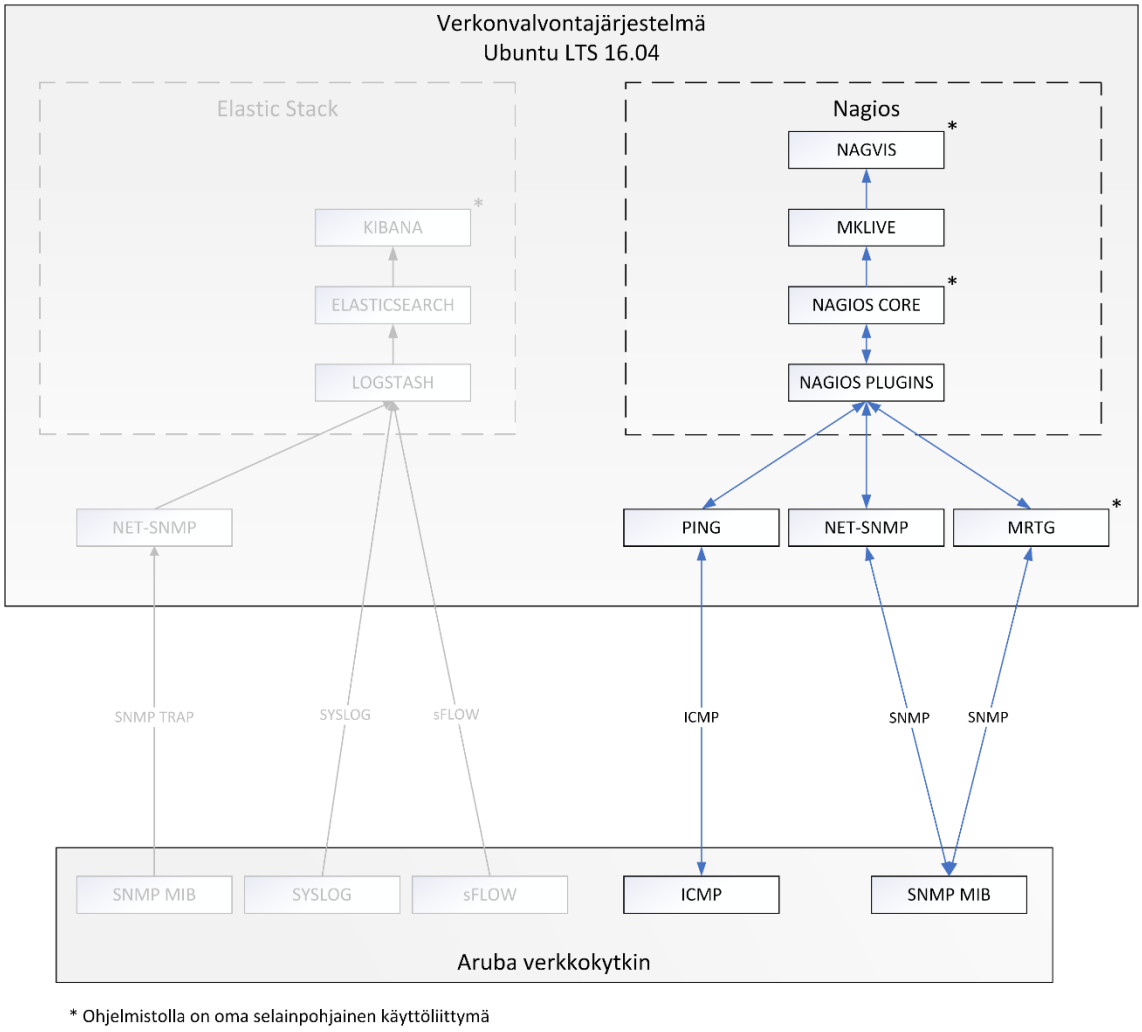


Kuva 33. *Kibanan selainpohjainen käyttöliittymä, jossa on visualisoituna kerättyjen Syslog-sanomien sisältö.*

5.2 Testaus ja tulokset

Toteutetun verkonvalvontajärjestelmän testaus on suoritettu useassa vaiheessa sekä erilaisissa ympäristöissä. Kehitysvaiheen testaus suoritettiin muutaman verkkokytkimen ja virtuaaliympäristöön asennetun Linux-käyttöjärjestelmän avulla. Verkonvalvontajärjestelmän kehitysvaiheen testaus suoritettiin toimistoympäristössä ja tuotantotestaus suoritettiin asiakkaan tuotantoympäristössä.

Toteutettu verkonvalvontakokonaisuus voidaan jakaa karkeasti kahteen osaan, joista ensimmäinen osa käsittelee aktiivista ja toinen passiivista valvontaa. Ensimmäinen osa toteutettiin alaluvun 5.1.1 mukaisena järjestelmänä, joka suorittaa kyselypohjasta valvontaa. Ensimmäinen osa otettiin käyttöön myös asiakkaan tuotantoympäristössä ja sitä on testattu käyttöönotosta lähtien seuraamalla säännöllisesti järjestelmän toimintaa. Tuotantoympäristössä käyttöönotettu osuus verkonvalvontajärjestelmästä on esitetty kuvassa 34.



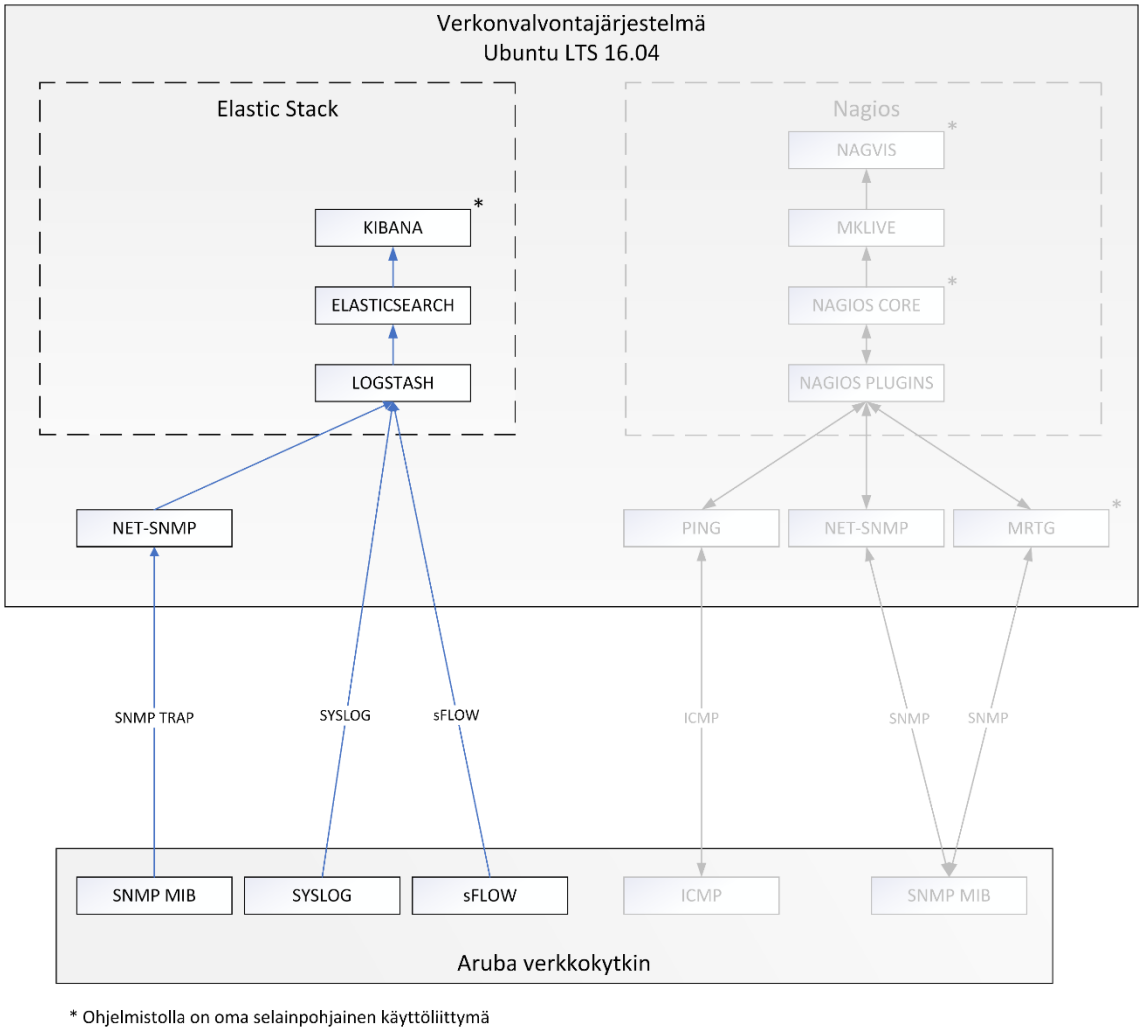
Kuva 34. *Aktiiviseen kyselypohjaiseen verkonvalvontaan perustuva osuus toteutetusta verkonvalvontakokonaisuudesta.*

Järjestelmän ensimmäisen osan tuotantoympäristön testauksen perusteella ollaan havaittu, että järjestelmä ei aiheuta haittaa verkon normaalille toiminnalle. Tietoa järjestelmän toiminnasta kerätään pidemmältä aikaväliltä, näin voidaan tulkita kerätyn tiedon avulla verkon normaalin toiminnan tunnusluvut. Näiden tunnuslukujen perusteella voidaan määritellä valvottavien kohteiden raja-arvot, esimerkiksi liikennemäärien raja-arvot. Tuotantotestaus on osoittanut järjestelmän toimivuuden verkon topologian ja verkkolaitteiden tilojen visualisoimisessa. Verkon toiminnan tila ja verkon rakenne on selkeästi ymmärrettävissä toteutetun järjestelmän avulla. Tuotantoympäristön verkko on toiminut testauksen aikana ilman suurempia ongelmia ja tämä on yksi syy siihen, miksi testausta on jatkettava edelleen tarkempien tulosten saavuttamiseksi.

Yksi testauksen aikana havaittu ongelma oli rikkoutunut Ethernet-kaapeli. Rikkoutunut kaapeli aiheutti verkkolaitteen rajapinnan tilan muutoksia, josta seurasi laitteiden välisiä yhteyskatkoja. Ongelma voitiin paikallistaa toteutetun järjestelmän avulla tietyn laitteen rajapintaan. Havaittiin kuitenkin ongelman paikantamisen olevan haasteellista, koska osa yhteyskatkoista oli niin lyhyitä, ettei niitä havaittu kyselypohjaisen valvonnan avul-

la. Ongelman paikantamisen jälkeen vian varsinaista juurisyytä jouduttiin selvittämään lisäksi verkkolaitteen sisäisestä lokista, joka edellytti laitteelle kirjautumista komentokehotteen avulla. Ongelman nopeampi selvittäminen olisi vaatinut tässä työssä toteutetun valvontakokonaisuuden toista osuutta. Toteutetun järjestelmän toinen osuus suorittaa passiivista valvontaa, eli lokitietojen ja virtaustallenteiden keruuta, muokkausta sekä visualisointia. Lokienkeruujärjestelmä avulla olisi ollut mahdollista kerätä Syslog ja SNMP Trap -sanomat jokaisesta rajapinnan tilan muutoksesta.

Järjestelmän toinen osa toteutettiin alaluvun 5.1.2 mukaisena järjestelmänä ja se suorittaa tapahtumapohjaista valvontaa. Järjestelmän toista osaa testattiin toimistoympäristössä. Testauksen perusteella havaittiin, että virtausteknologioissa on potentiaalia liikenteen seurannassa ja verkon laitteiden suorituskyvyn valvonnassa. Virtausteknologioita pitää testata niiden hyödyllisyyden varmistamiseksi ja todellisten tulosten saamiseksi vielä tuotantoympäristössä. Kuvassa 35 on esitetty järjestelmän toinen osuus.



Kuva 35. *Passiiviseen tapahtumapohjaiseen verkonvalvontaan perustuva osuus toteutetusta verkonvalvontakokonaisuudesta.*

Järjestelmän testauksessa käytettiin Aruba-merkkisten verkkokytkimien lisäksi Siemensin Scalance-sarjan kytkimiä sekä S7-1200 sarjan ohjelmoitavaa logiikkaa. Scalance-sarjan kytkimistä löytyy tuki Profinet-liikenteelle. Sarjasta on saatavissa myös malleja, jotka täyttävät Profinet-verkkoliikenteen kovimmat reaaliaikavaatimukset. Kun tarvitaan tukea Profinet-protokollalle, joudutaan Scalancen tapauksessa luopumaan informaatioteknologian tarpeisiin suunnatuista menetelmistä, kuten Syslog-sanomista sekä virtausteknologioista. Testattaessa järjestelmän toista osaa, todettiin kuitenkin Scalancen tukeman SNMP-protokollan suoriutuvan passiivisen verkonvalvonnan tehtävistä SNMP Trap -sanomien avulla. Ohjelmoitavan logiikan osalta SNMP-protokollan MIB-tietovarasto oli hyvin rajoittunut, eikä laite tukenut lainkaan Syslog tai SNMP Trap -sanomia.

5.3 Jatkotutkimus ja -kehitys

Nagioksen käyttöönotto oli ensimmäisellä kerralla hieman monimutkainen ja työläs. Tämä johtui pääasiassa Nagioksen konfiguroinnin monipuolisuudesta, laajuudesta sekä hajautetuista ja tekstipohjaisista konfigurointitiedostoista. Toteutuksen alkuvaiheessa täytyi tutustua Nagioksen dokumentaatioon huolellisesti ja tunnistaa tarvittava muunneltavuus. Nagioksen ympärillä on hyvä ja aktiivinen kehitysyhteisö ja Nagios tarjoaa laajan tukimateriaalin ongelmien ratkaisemisen tueksi.

Nagioksen käyttöönottoa uusissa valvontajärjestelmissä voidaan helpottaa toteuttamalla määrittelydokumentti, esimerkiksi Excel-taulukkolaskentaohjelmistolla. Määrittelydokumentin avulla voidaan muodostaa automaattisesti Nagiosen konfiguroimisessa tarvittavat laitekohtaiset määrittelytiedostot. Määrittelydokumentin avulla pystytään nopeuttamaan uuden kokonaisuuden käyttöönottoa sekä vähentämään inhimillisiä virheitä tekstipohjaisia määrittelytiedostoja luotaessa. Määrittelydokumentin avulla on mahdollista vähentää käyttöönottajan tarvetta perehtyä Nagioksen dokumentaatioon. Tämän avulla voidaan keventää käyttöönottoprosessia merkittävästi. Määrittelydokumentin avulla on mahdollista tuottaa myös Nagvis-ohjelmiston karttapohjien määrittelytiedostot automaattisesti.

Toinen Nagioksen kehittämiskohde on hälytykset. Sähköpostihälytyksiä ei ole mahdollista toimittaa vastaanottajille yhteyden ollessa estynyt automaatiojärjestelmästä ulkoverkkoon. Hälytysten toimittaminen asianomaisille henkilöille, myös tällaisessa häiriötilanteessa, vaatisi GSM (Global System for Mobile Communications) -jatkohälytyksien käyttöönottamista.

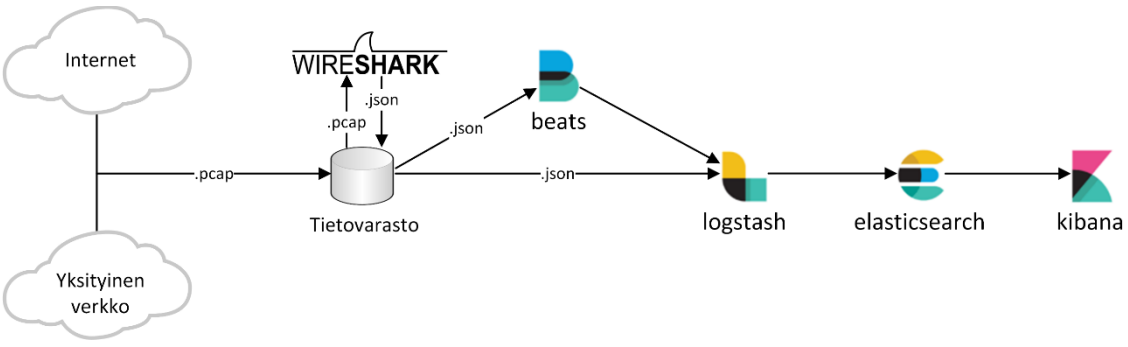
RRDtool-ohjelmisto on uudelleensuunnittelu MRTG:n grafiikan ja lokientallennus toiminnallisuudesta. RRDtool ei kuitenkaan suorita SNMP-objektien kyselyitä MRTG:n tavoin. Näiden kahden ohjelmiston yhteistyötä voidaan hyödyntää esimerkiksi niin, että MRTG huolehtii SNMP-objektien keräämisestä ja RRDtool kerätyn tiedon tallentamisesta ja visualisoinnista. RRDtool tallentaa tiedot RRD-tietokantaan. RRD-tietokanta

tarjoaa nopeamman pääsyn tietoihin sekä tehokkaamman tavan toteuttaa visualisointi. MRTG luo grafiikat jokaisen SNMP-objektien kyselyn yhteydessä, kun RRDtool luo grafiikat vain pyydettäessä kyseistä grafiikkaa, esimerkiksi selaimen käyttöliittymällä. RRDtoolin tapa luoda grafiikat tarvittaessa vähentää palvelimen kuormitusta. Kuormituksen vähentäminen on merkittävää kun valvottavia verkkolaitteita ja niiden arvoja luetaan suuria määriä. [73]

Jatkossa on syytä miettiä MRTG:n ja RRDtoolin yhteistyön hyödyntämistä. Koska kuvaajia piirretään paljon, RRDtoolin käytöllä voidaan vähentää kuvaajien piirtämisestä johtuvaa kuormitusta sekä tehostaa tiedonhakua. RRD-tietokantaan tallennetut tiedot ja niistä muodostetut kuvaajat voidaan jatkossa upottaa suoraan Nagioksen käyttöliittymään. Kuvaajien integroiminen Nagiokseen helpottaa valvottavien arvojen pitkäaikaista seurantaa, eikä erilliselle MRTG:n tarjoamalle käyttöliittymäsivulle ole enää tarvetta.

Jos SNMP-protokollan ja virtausteknologioiden avulla ei saavuteta riittävää tasoa automaatioverkon profiileiden valvonnassa, esimerkiksi yhteensopivuusongelmien vuoksi, vaihtoehtoisesti voidaan tutkia pakettikaappauksen soveltuvuutta kyseiseen tehtävään. Yhteensopivuusongelmalla tarkoitetaan tässä yhteydessä automaatiolaitteen puutteellista SNMP-protokollatukea tai esimerkiksi automaatioverkon profiilin reaaliaikavaatimuksesta johtuvan erikoislaitteen käyttöä, joka ei tue virtausteknologioita. Automaatioverkon laitteiden verkonvalvontaominaisuuksien rajoittuneisuus saattaa johtaa pakettikaappauksen käyttöönoton suunnitteluun. Yksi mahdollinen menetelmä toteuttaa pakettien kaappaus on käyttää Wiresharkin pakettienkaappaus- ja analysointityökalua.

Wiresharkin toteuttaman Tshark-sovelluksen avulla voidaan kaapata paketteja verkosta reaaliaikaisesti tai analysoida kaappaustallenteita. Tshark tukee reaaliaikaisten pakettikaappausten välittämistä suoraan Elastic Stack -ohjelmistopinolle JSON-dokumentteina. [74] Wireshark tukee myös virallisesti useita automaatioverkon profiileita, mukaan lukien Profinet, EtherNet/IP sekä Modbus/TCP [75]. Kuvassa 36 on esitetty Wiresharkin ja Elastic Stackin -ohjelmistopinon yhteistyö pakettikaappausten analysoimiseksi.



Kuva 36. *Wiresharkin ja Elastic Stackin -ohjelmistopinon yhteistyö pakettikaappauksien analysoimiseksi. Mukailtu lähteestä [74].*

Tietoturvan osuus tämän tutkimuksen puitteissa jäi hyvin vähäiseksi. Tutkimuksessa koostettua verkonvalvontakokonaisuutta pitää tarkastella vielä erikseen tietoturvanäkökulmien kannalta. Tietoturvaa tulee tarkastella liikenteen salauksen sekä muiden tietoturvaominaisuuksien osalta. Tässä tutkimuksessa käytetyt ohjelmistot, kuten Nagios ja Elastic Stack, tarjoavat itsessään tietoturvaominaisuuksia. Nagis keskittyy pääasiassa ohjelmisto tasolla käyttäjäpohjaiseen tietoturvaan. Elastic Stack tarjoaa menetelmiä käyttäjäpohjaiseen tietoturvaan sekä hajautettujen klusteritoteutusten välisen kommunikoinnin salaamiseen. Verkonvalvontaan soveltuvien protokollien tietoturvatarkastelu tulisi myös toteuttaa. Protokollat tarjoavat itsessään tietoturvaominaisuuksia tai mahdollistaa tietoturvan toteuttamisen käyttämällä turvallista kuljetuskerroksen protokollaa.

Tietoturvan määrittelyssä pitäisi lähteä liikkeelle käyttökohteiden tarkastelusta ja kohteiden riskien arvioinnista. Tietoturva-analyysin sekä riskianalyysin avulla voidaan määritellä tietoturvalta vaadittava taso. Tason määrittämisen jälkeen on mahdollista tarkastella menetelmiä, joiden avulla vaadittava taso pystytään saavuttamaan.

6 YHTEENVETO

Tämän työn tavoitteena oli tutkia Ethernet-pohjaisten teollisuusautomaatioverkkojen kunnonvalvontaa sekä selvittää informaatioteknologiassa käytettyjen menetelmien soveltuvuutta kyseiseen tarkoitukseen. Tutkimuksessa hyödynnettiin avoimen lähdekoodin työkaluja, joiden avulla koostettiin automaatioverkkoon soveltuva verkonvalvontakokonaisuus. Tässä yhteenvedossa käydään läpi tutkimuskysymykset sekä näihin kysymyksiin löydetyt vastaukset.

Tutkimuksen aihealue ja sen laajuus aiheuttivat haasteita työn rajauksessa. Toisena haasteena oli verkonvalvontaa ja erityisesti teollisuuden verkkojen valvontaa koskevien lähteiden löytäminen. Vaikeuksia lähteiden löytämisessä aiheutti englanninkielinen kattotermi verkonhallinnalle (network management), joka kattaa myös verkonvalvonnan (network monitoring). Edellä mainittuja termejä käytetään ristiin keskenään kirjallisuuslähteissä. Näiden lisäksi on olemassa myös muita termejä, joilla viitataan samoihin asiayhteyksiin. Kirjoittajasta riippuu mitä termiä käytetään ja mitä sillä milloinkin tarkoitetaan.

1. *Mitkä ovat informaatioteknologiassa yleisesti hyväksytyt verkonhallinta ja -valvonta periaatteet sekä mitkä ovat niiden yleiset käytännöt?*

Lähtökohtana verkonhallinnan ja valvonnan periaatteille ovat erilaiset verkonhallinnan ja -valvonnan viitemallit. Viitemallit tarjoavat kehyksen verkonhallinnan ja -valvonnan toteuttamiseksi verkkolaitetasolta aina yrityksen liiketoiminnan hallintaan asti. Tämän tutkimuksen osalta keskityttiin TMN-viitemallin kolmeen alimpaan kerrokseen, joita ovat elementit, elementtienhallinta sekä verkonhallinta. TMN-viitemalli sitoo verkonvalvonnan osa-alueet yhteen ja tuo ne osaksi yrityksen liiketoimintaa.

Yksittäisten verkkolaitteiden riippuvuutta yrityksen varsinaiselle liiketoiminnalle voi olla vaikea nähdä. Riippuvuus on kuitenkin olemassa ja laitteet ovat oleellinen osa liiketoimintaa, jos liiketoiminnan kannalta kriittisiä toimintoja tuotetaan verkon avulla. Vaikka automaatioalan yritys ei yleensä tuota asiakkaille varsinaisia verkkopalveluita, verkko on tärkeässä asemassa automaation ohjausjärjestelmissä. Automaatioverkkoa tarvitaan prosessin ohjauksessa, prosessin ja ihmisten välisessä vuorovaikutuksessa sekä raportoitaessa prosessin ja liiketoiminnan kannalta merkityksellistä tietoja. Näin voidaan nähdä automaatioverkon yksittäisten verkkolaitteiden merkitys yrityksen liiketoiminnalle.

TMN-viitemalliin sidoksissa olevan FCAPS-toimintamallin avulla pystytään jakamaan hallittavien kokonaisuuksien toiminnallisia vaatimuksia pienempiin osa-alueisiin ja pai-

nottamaan näitä osa-alueita suhteutettuna yrityksen tarpeisiin. Verkonhallinnan jako hallintaan ja valvontaan esitettiin alaluvussa 4.3. Alaluvun kuvan 16 esittämä jako on hieman vanhanaikainen. Jako hallintaan ja valvontaan voidaan ajatella myös niin, että yritys määrittelee suhteet hallinnan ja valvonnan painopisteille. Yritys määrittelee ensimmäiseksi mitkä FCAPS-toimintamallin mukaiset osa-alueet toteutetaan, minkä jälkeen määritellään valittujen osa-alueiden hallinnan ja valvonnan välinen suhde. Tässä tutkimuksessa toteutettu verkonvalvontakokonaisuus painottui FCAPS-toimintamallin vianhallinnan sekä suorituskyvynhallinnan osa-alueisiin. Näiden kahden osa-alueen kohdalla keskityttiin ainoastaan valvontaan.

2. *Mitkä ovat verkonvalvonnan menetelmät ja kuinka ne soveltuvat teollisuusautomaation Ethernet-pohjaisiin tietoliikenneverkkoihin?*

Verkonvalvonnan menetelmät ovat tapoja, joiden avulla ensimmäisen tutkimuskysymyksen osa-alueita ja vaatimuksia voidaan käytännössä toteuttaa. Näitä menetelmiä on useita ja vain murto osaa käsiteltiin tämän tutkimuksen puitteissa. Tähän tutkimukseen valikoituneet menetelmät ovat laajasti hyväksyttyjä sekä tuettuja informaatioteknologian verkkolaitteissa. Valitut menetelmät olivat verkonhallintaprotokolla SNMP, ICMP-protokolla tiedonsiirron diagnostiikan lähettämiseksi sekä järjestelmäsanomien lähettämisessä käytettävä Syslog-protokolla. Näiden lisäksi perehdyttiin kolmeen virtausteknologiaan, joita olivat NetFlow, IPFIX sekä sFlow. Verkonvalvonnan menetelmiä käsiteltiin tarkemmin alaluvussa 4.5.

Periaatteen tasolla ei ole olemassa rajoitteita sille, etteikö informaatioteknologiassa käytössä olevia menetelmiä voi soveltaa automaatioverkoissa. Käytännössä asia on hieman toisenlainen. Suurin ongelma useimpien teknologioiden osalta on niiden hidas rantautuminen automaatioverkon laite tasolle. Vanhimmat ”force major” teknologiat löytyvät automaatioverkon laitetasolta, kuten kenttälaitetasolta, mutta niiden ominaisuuksia on yleensä rajoitettu.

Kun automaatioverkossa kulkevien sanomien reaaliaikavaatimukset kasvavat, edellä kuvailtu rajoittuneisuus lisääntyy huomattavasti. TCP/IP-viitemallin mukainen rakenne alkaa muokkaantumaan aina alimmille kerroksille saakka, mikä johtaa erityistarkoituksiin suunniteltujen verkkolaitteiden käytön tarpeeseen. Samalla markkinoilla olevien ja käyttötarkoitukseen soveltuvien laitteiden määrä vähenee murto-osaan.

Verkonvalvonta on kuitenkin mahdollista toteuttaa pääsääntöisesti informaatioteknologian menetelmillä, kun reaaliaikavaatimukset pysyvät kohtuullisina. Tällöin voidaan käyttää informaatioteknologian käyttöön suunniteltuja laitteita. Näiden laitteiden käyttö mahdollistaa esimerkiksi virtastallenteiden keräämisen automaatioverkosta.

3. *Soveltuvatko avoimen lähdekoodin työkalut yrityksen automaatioverkkovalvonnan tarpeisiin?*

Tutkimukseen oli valittu kaksi avoimen lähdekoodin sovelluskokonaisuutta, joiden pohjalta toteutusta lähdettiin pohtimaan. Nämä sovellukset olivat Nagios Core sekä Elastic Stack -ohjelmistopino. Valinnat perustuvat asiakastarpeeseen sekä yrityksen aikaisempaan kokemukseen ja tutkimukseen. Sovellukset myös mahdollistavat tilaajan olemassa sekä kehitteillä olevien ominaisuuksien integroimisen tässä tutkimuksessa koostettuun kokonaisuuteen. Nämä sovellukset eivät itsessään täyttäneet kaikkia valvontakokonaisuudelle asetettuja vaatimuksia, vaan tarvitsivat lisäksi erilaisia sovelluksia ja lisäosia toimivan kokonaisuuden saavuttamiseksi. Tarvittavista lisäosista pääosassa olivat järjestelmän visualisoinnissa käytetty Nagvis ja liikennemäärien keruuseen sekä graafisten kuvaajien piirtämiseen käytetty MRTG-sovellus. Verkonvalvonnan toteutus on käsitelty luvussa 5, jonka kuvassa 25 on esitetty tarkemmin kokonaisuuden arkkitehtuuri, käytetyt sovellukset ja lisäosat.

Vaatimuksena tässä tutkimuksessa koostetulle verkonvalvontakokonaisuudelle oli valmistajariippumattomuus sekä laajennettavuus tulevaisuuden muuttuvia tarpeita varten. Yhtenä vaatimuksena oli myös toteuttaa visuaalisesti havainnollinen järjestelmä, jonka avulla verkon ongelman sijainti voidaan paikantaa ja esittää järjestelmän ylläpitäjälle. Tutkimuksessa onnistuttiin koostamaan verkonvalvontakokonaisuus, joka saatiin toteutettua sille määriteltyjen vaatimusten mukaisesti. Sovelluskokonaisuus täyttää kysely- sekä tapahtumapohjaisen verkonvalvonnan tarpeet. Sovelluskokonaisuuden avulla pystyttiin vastaamaan FCAPS-toimintamallin mukaisten vikojenvalvonnan ja suorituskyvyn valvonnan osa-alueisiin. Kokonaisuus kaipaa vielä jatkotutkimusta sekä -kehitystä, että siitä saadaan valmis, saumattomasti yhteen toimiva kokonaisuus. Tutkimuksen aikana esille nousseita jatkotutkimus ja -kehitys kohteita on käsitelty alaluvussa 5.3. Löydetyistä jatkotutkimus ja -kehitys kohteista tärkeimpiä olivat Nagioksen käyttöönoton tehostaminen Excel-pohjaisen määrittelydokumentin avulla, MRTG-sovelluksen keräämän tiedon säilyttäminen ja graafisten kuvaajien tuottaminen tehokkaammin RRDtool-sovelluksen avulla sekä toteutetun verkonvalvontakokonaisuuden arvioiminen turvallisuusnäkökulmista.

Toteutetun verkonvalvontajärjestelmän avulla voidaan tukea asiakasyrityksen liiketoimintaa sekä varmistaa verkon toiminnan jatkuvuus. Järjestelmä mahdollistaa nopean reagoinnin vikatilanteessa ja sen avulla pystytään paikantamaan vika oikeaan sijaintiin. Järjestelmän avulla voidaan tunnistaa verkossa alkavia ongelmia ja mahdollisesti välttää verkon vikaantuminen kokonaan. Kerättyjen tietojen ja ennakoivan analyysin avulla voidaan suunnitella esimerkiksi verkkoon tarvittavat huoltotoimenpiteet sekä kapasiteetinmuutokset. Ennakoivalla toiminnalla voidaan välttää verkosta aiheutuvat suunnittelemattomat tuotantoa häiritsevät katkokset.

LÄHTEET

[1] J. Ding, Advances in Network Management, 1st ed. CRC Press Taylor & Francis Group, Boca Raton, Florida, USA, 2009, 390 p.

[2] O.C. Ibe, Fundamentals of Data Communication Networks, John Wiley & Sons, Incorporated, Newark, New Jersey, USA, 2017, 333 p.

[3] H. Jaakohuhta, Lähiverkot: Ethernet, 4th. ed. IT Press, Helsinki, 2005, 380 p.

[4] J.F. Kurose, K.W. Ross, Computer Networking: a Top-Down Approach, 5th ed. Pearson, New York, USA, 2010, 888 p.

[5] M. Puska, Lähiverkkojen Tekniikka: Pro Training, 2nd ed. Suomen atk-kustannus, Helsinki, 2000, 348 p.

[6] C. Spurgeon, R. Torkkeli, Ethernet: Tehokäyttäjän Opas, Suomen atk-kustannus, Helsinki, 2001, 531 p.

[7] E.D. Knapp, J.T. Langill, Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, 2nd ed. Syngress Media Incorporated, Waltham, Massachusetts, USA, 2015, 460 p.

[8] J.T.J. Penttinen, The Telecommunications Handbook: Engineering Guidelines for Fixed, Mobile and Satellite Systems, 1st ed. John Wiley & Sons Inc, West Sussex, UK, 2015, 957 p.

[9] P. Zhang, Advanced Industrial Control Technology, 1st ed. Elsevier Inc, Kidlington, UK, 2010, 842 p.

[10] B.G. Liptak, Instrument Engineers' Handbook: Process Software and Digital Networks, Volume 3, 4th ed. CRC Press Taylor & Francis Group, Boca Raton, Florida, USA, 2011, 1079 p.

[11] R. Zurawski, Industrial Communication Technology Handbook, 2nd ed. CRC Press Taylor & Francis Group, Boca Raton, Florida, USA, 2014, 1757 p.

[12] B. Galloway, G.P. Hancke, Introduction to Industrial Control Networks, IEEE Communications Surveys & Tutorials, Vol. 15, Iss. 2, 2013, pp. 860-880.

[13] J.R. Moyne, D.M. Tilbury, The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data, Proceedings of the IEEE, Vol. 95, Iss. 1, 2007, pp. 29-47.

[14] J. Seppälä, M. Salmenperä, Towards dependable automation, in: M. Lehto, P. Neittaanmäki (ed.), Cyber Security: Analytics, Technology and Automation, Vol. 78, Springer International Publishing, 2015, pp. 229-249.

[15] J. Wang, Real-Time Embedded Systems, John Wiley & Sons Inc, Hoboken, New Jersey, USA, 2017, 310 p.

[16] Y. Li, D. Li, W. Cui, R. Zhang, Research based on OSI model, 2011 IEEE 3rd International Conference on Communication Software and Networks, Xi'an, China, May 27-29, 2011, IEEE, Piscataway, New Jersey, USA, pp. 554-557.

[17] D.E. Comer, Computer networks and Internets, 5th ed. Pearson Education International, Upper Saddle River, New Jersey, USA, 2009, 600 p.

[18] R. Zurawski, The Industrial Communication Technology Handbook, 1st ed. CCR Press Taylor & Francis Group, Boca Raton, Florida, USA, 2005, 1756 p.

[19] S.K. Sen, Fieldbus and Networking in Process Automation, 1st ed. CRC Press Taylor & Francis Group, Bosa Roca, USA, 2014, 439 p.

[20] M. Hakala, M. Vainio, Tietoverkon Rakentaminen, 2nd ed. Docendo, Jyväskylä, 2005, 428 p.

[21] K. Kaario, TCP/IP-verkot, Docendo, Jyväskylä, 2002, 396 p.

[22] R. Pigan, M. Metter, Automating with Profinet, 2nd ed. Publicis Publishing, Germany, 2008, 462 p.

[23] M. Rostan Industrial Ethernet Technologies, EtherCAT Technology Group, verkkosivu. Saatavissa (viitattu 13.07.2018): http://www.ethercat.org/pdf/english/Industrial_Ethernet_Technologies.pdf.

[24] Profinet IO Conformance Classes, Profibus and Profinet International, verkkosivu. Saatavissa (viitattu 16.10.2018): https://www.profibus.com/index.php?eID=dumpFile&t=f&f=44266&token=1d81c134b224334c62d388673080f12267581e62.

[25] A Guide for EtherNet/IP™ Developers, ODVA, verkkosivu. Saatavissa (viitattu 04.10.2018): https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00213R0_EtherNetIP_Developers_Guide.pdf.

[26] Ethernet/IP, ODVA, verkkosivu. Saatavissa (viitattu 04.10.2018): https://www.odva.org/Technology-Standards/EtherNet-IP/Overview.

[27] V. Schiffer The Common Industrial Protocol (CIP™) and the Family of CIP Networks, ODVA, verkkosivu. Saatavissa (viitattu 04.10.2018): https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf.

[28] EtherNet/IP Performance, Rockwell Automation, verkkosivu. Saatavissa (viitattu 04.10.2018): http://www.politecnica.pucrs.br/professores/tergolina/Redes_e_Protocolos_Industriais/LITERATURA_ADICIONAL_-_EtherNetIP_enet-ap001_-en-p.pdf.

[29] Modbus Application Protocol Specification 1.1a, Modbus Organization, verkkosivu. Saatavissa (viitattu 03.10.2018): http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1a.pdf.

[30] Modbus, Modbus Organization, verkkosivu. Saatavissa (viitattu 03.02.2018): http://www.modbus.org/.

[31] A. Clemm, Network Management Fundamentals, 1st ed. Cisco Press, Indianapolis, USA, 2007, 510 p.

[32] W. Goralski, The Illustrated Network: How TCP/IP Works in a Modern Network, 2nd ed. Elsevier Inc, Cambridge, Massachusetts, USA, 2017, 936 p.

[33] A. Kwiecień, K. Opielka, Management of Industrial Networks Based on the FCAPS Guidelines, Computer Networks 19th International Conference, CN 2012, Szczyrk, Poland, June 19-23, 2012, Springer, Heidelberg, DE, pp. 280-288.

[34] D. Mauro, K. Schmidt, Essential SNMP: Help for System and Network Administrators, 2nd ed. O'Reilly Media, 2005, 460 p.

[35] J. Sathyan, Fundamentals of EMS, NMS and OSS/BSS, 1st ed. CRC Press Taylor & Francis Group, Boca Raton, Florida, USA, 2010, 588 p.

[36] Subramanian Mani, Network Management: Principles and Practices, 2nd ed. Pearson, India, 2010, 726 p.

[37] J.D. McCabe, Network Analysis, Architecture, and Design, 3rd ed. Elsevier, Burlington, Massachusetts, USA, 2007, 495 p.

[38] A. Kwiecień, P. Gaj, P. Stera, Computer Networks: 19th International Conference, CN 2012, Szczyrk, Poland, June 19-23, 2012. Proceedings, Springer, Heidelberg, DE, 2012, 481 p.

[39] D. Kidston Distributed Network Management, Communications Research Centre Canada, verkkosivu. Saatavissa (viitattu 18.7.2018): http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.896.9723&rep=rep1&type=pdf.

[40] J. Bergstra, M. Burgess, M. Burgess, Handbook of Network and System Administration, Elsevier, Oxford, 2008, 1028 p.

[41] H. Ji, B. Zhang, G. Li, X. Gao, Y. Li, Challenges to the New Network Management Protocol: NETCONF, 2009 First International Workshop on Education Technology and Computer Science, Wuhan, Hubei, China, March 7-8, 2009, IEEE, Piscataway, New Jersey, USA, pp. 832-836.

[42] J. Schoenwaelder RFC 3535 Overview of the 2002 IAB Network Management Workshop, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 08.08.2018): https://tools.ietf.org/html/rfc3535#section-2.1.

[43] W. Odom, CCNA Routing and Switching ICND2 200-105 Official Cert Guide, 1st ed. Cisco Press, Indianapolis, USA, 2016, 992 p.

[44] G. Liu, N. Neufeld, Management of the LHCb Readout Network, IEEE Transactions on Nuclear Science, Vol. 57, Iss. 2, 2010, pp. 715-720.

[45] D. Oancea, Structure of Management Information in SNMP, Analele Universităţii "Dunărea de Jos" Galaţi: Fascicula III, Vol. 2003, Iss. 1, 2003, pp. 87-90. https://doaj.org/article/0431aa3c2b7d484ab8751b58ed694617.

[46] J. Postel RFC 792 Internet Control Message Protocol, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 17.07.2018): https://tools.ietf.org/html/rfc792.

[47] R. Gerhards RFC 5424 The Syslog Protocol, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 17.07.2018): https://tools.ietf.org/html/rfc5424.

[48] A. Okmianski RFC 5426 Transmission of Syslog Messages over UDP, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 17.07.2018): https://tools.ietf.org/html/rfc5426.

[49] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras, Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX, IEEE Communications Surveys & Tutorials, Vol. 16, Iss. 4, 2014, pp. 2037-2064.

[50] J. Quittek, T. Zseby, B. Claise & S. Zander, RFC 3917 Requirements for IP Flow Information Export (IPFIX), Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 23.07.2018): https://tools.ietf.org/html/rfc3917.

[51] B. Claise, B. Trammell & P. Aitken, RFC 7011 Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 25.04.2018): https://tools.ietf.org/html/rfc7011.

[52] B. Li, J. Springer, G. Bebis, M. Hadi Gunes, A Survey of Network Flow Applications, Journal of Network and Computer Applications, Vol. 36, Iss. 2, 2013, pp. 567-581.

[53] G. Sadasivan, N. Brownlee, B. Claise & J. Quittek, RFC 5470 Architecture for IP Flow Information Export, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 23.07.2018): https://tools.ietf.org/html/rfc5470.

[54] B. Claise RFC 3954 Cisco Systems NetFlow Services Export Version 9, Internet Engineering Task Force, verkkosivu. Saatavissa (viitattu 20.04.2017): https://www.ietf.org/rfc/rfc3954.txt.

[55] Kimball John NetFlow vs sFlow: What's the Difference and Which is Better? Comparitech, verkkosivu. Saatavissa (viitattu 25.8.2018): https://www.comparitech.com/net-admin/netflow-vs-sflow/#gref.

[56] NetFlow Version 9 Flow-Record Format, Cisco, verkkosivu. Saatavissa (viitattu 18.10.2018): https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html.

[57] sFlow Technology, InMon Corporation, verkkosivu. Saatavissa (viitattu 24.07.2018): https://inmon.com/technology/.

[58] P. Phaal, M. Lavine, sFlow Version 5, sflow.org, verkkosivu. Saatavissa (viitattu 28.04.2018): https://sflow.org/sflow_version_5.txt.

[59] Satellar XT 5RC, Satel Oy, verkkosivu. Saatavissa (viitattu 18.10.2018): https://www.satel.com/fi/tuotteet/radiomodeemit/satellar-xt-5rc/.

[60] Satellar Digital System Part II: Central Unit User Guide Version 1.8, SATEL Oy, verkkosivu. Saatavissa (viitattu 18.10.2018): https://www.satel.com/wp-content/uploads/2017/08/SATELLAR_CU_V1_8.pdf.

[61] M. Guthrie, Instant Nagios Starter, 1st ed. Packt Publishing, Instant Nagios Starter, UK, 2013, 47 p.

[62] W. Kocjan, Learning Nagios 4, 1st ed. Packt Publishing, Birmingham, UK, 2014, 209 p.

[63] NagVis 1.9 Documentation, NagVis Project Team, verkkosivu. Saatavissa (viitattu 03.03.2018): http://docs.nagvis.org/1.9/en_US/index.html.

[64] MK Livestatus, Mathias Kettner GmbH, verkkosivu. Saatavissa (viitattu 04.03.2018): https://mathias-kettner.de/checkmk_livestatus.html.

[65] T. Oetiker MRTG - Documentation, Oetiker+Partner AG, verkkosivu. Saatavissa (viitattu 12.09.2018): https://oss.oetiker.ch/mrtg/doc/mrtg.en.html.

[66] B. Dixit, Elasticsearch: A Complete Guide, 1st ed. Packt Publishing, Birmingham, UK, 2017, 826 p.

[67] Beats, Elastic, verkkosivu. Saatavissa (viitattu 11.08.2018): https://www.elastic.co/products/beats.

[68] D. Berman The Complete Guide to the ELK Stack – 2018, Logz.io, verkkosivu. Saatavissa (viitattu 04.09.2018): https://logz.io/learn/complete-guide-elk-stack/.

[69] Filebeat Reference, Elastic, verkkosivu. Saatavissa (viitattu 13.08.2018): https://www.elastic.co/guide/en/beats/filebeat/current/.

[70] Logstash Reference, Elastic, verkkosivu. Saatavissa (viitattu 15.08.2018): https://www.elastic.co/guide/en/logstash/current/.

[71] Logstash, Elastic, verkkosivu. Saatavissa (viitattu 14.08.2018): https://www.elastic.co/products/logstash.

[72] Elasticsearch Reference, Elastic, verkkosivu. Saatavissa (viitattu 15.08.2018): https://www.elastic.co/guide/en/elasticsearch/reference/current/.

[73] Denenberg Adam Advanced SNMP Monitoring with RRDTool, UBM Technology Group, verkkosivu. Saatavissa (viitattu 14.09.2018): http://www.drdobbs.com/advanced-snmp-monitoring-with-rrdtool/199101665.

[74] C. Wurm Analyzing Network Packets with Wireshark, Elasticsearch, and Kibana, Elastic, verkkosivu. Saatavissa (viitattu 14.08.2018): https://www.elastic.co/blog/analyzing-network-packets-with-wireshark-elasticsearch-and-kibana.

[75] G. Harris Fieldbus Protocol Family, Wireshark Foundation, verkkosivu. Saatavissa (viitattu 20.09.2018): https://wiki.wireshark.org/FieldbusProtocolFamily.

LIITE A: STANDARDIN IEC 61158 MÄÄRITTELEMÄT AUTOMAATIOVERKON PROFIILIT

Table 1. *Structure of IEC 61158. [11]*

IEC 61158 Parts	Content
IEC 61158-1	Introduction
IEC 61158-2-x	PHL: Physical Layer
IEC 61158-3-x	DLL: Data Link Layer Service
IEC 61158-4-x	DLL: Data Link Layer Protocols
IEC 61158-5-x	AL: Application Layer Services
IEC 61158-6-x	AL: Application Layers Protocol
IEC 61158-7-x	Network Management

Note: x indicates the relate CPF.

Table 2. *Standards related with profiles. [11]*

Standard	Description
IEC 61784-1	Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems
IEC 61784-2	Additional profiles for ISO/IEC 8802-3 based communication networks in RT applications
IEC 61784-3-x	Profiles for functional safe communications in industrial networks
IEC 61784-4-x	Profiles for secure communications in industrial networks
IEC 61784-5-x	Installation profiles for communication networks in industrial control systems

Note: x indicates the relate CPF.

Table 3. *List of communication profile families. [11]*

Communication Profile Family	Name
CPF1	FOUNDATIONTM Fieldbus
CPF2	ControlNetTM
CPF3	PROFIBUS
CPF4	P-NET®
CPF5	WorldFIP®
CPF6	INTERBUS®
CPF7	SwiftNet
CPF8	CC-Link
CPF9	HART®
CPF10	VNET/IP
CPF11	TCnet
CPF12	EtherCAT®
CPF13	EPL
CPF14	EPA
CPF15	MODBUS® - RTPS
CPF16	SERCOS
CPF17	RAPIEnet
CPF18	SafetyNET pTM
CPF19	MECHATROLINK

Table 4. *Relation between CPF, CP, and type of protocol. [11]*

IEC 61784			IEC 61158 Services and Protocols			
Family	**Part 1**	**Part 2**	**PHL**	**DLL**	**AL**	**Brand Names**
Family 1						*Foundation Fieldbus (FF)*
	Profile 1/1		Type 1	Type1	Type 9	Foundation-H1
	Profile 1/2		8802-3	TCP/UDP/IP	Type 5	Foundation-HSE
	Profile 1/3		Type 1	Type 1	Type 9	Foundation-H2
Family 2						*CIP*
	Profile 2/1		Type 2	Type 2	Type 2	ControlNet
	Profile 2/2	Profile 2/2	8802-3	TCP/UDP/IP	Type 2	EtherNet/IP
		Profile 2/2.1	8802-3	TCP/UDP/IP	Type 2	EtherNet/IP with time synchronization
	Profile 2/3		Type 2	Type 2	Type 2	DeviceNet
Family 3						*PROFIBUS & PROFINET*
	Profile 3/1		Type 3	Type 3	Type 3	PROFIBUS DP
	Profile 3/2		Type 1	Type 3	Type 3	PROFIBUS PA
	Profile 3/3		8802-3	TCP/IP	Type 10	PROFINET CBA
		Profile 3/4	8802-3	Type 10	Type 10	PROFINET IO Class A
		Profile 3/5	8802-3	Type 10	Type 10	PROFINET IO Class B
		Profile 3/6	8802-3	Type 10	Type 10	PROFINET IO Class C
Family 4						*P-NET*
	Profile 4/1		Type 4	Type 4	Type 4	P-NET RS-485
	Profile 4/2		Type 4	Type 4	Type 4	P-NET RS-232 (removed 2013)
		Profile 4/3				P-NET on IP
Family 5						*WorldFIP*
	Profile 5/1		Type 1	Type 7	Type 7	WorldFIP (MPS, MCS)
	Profile 5/2		Type 1	Type 7	Type 7	WorldFIP (MPS, MCS, SubMMS)
	Profile 5/3		Type 1	Type 7	Type 7	WorldFIP (MPS)
Family 6						*INTERBUS*
	Profile 6/1		Type 8	Type 8	Type 8	INTERBUS
	Profile 6/2		Type 8	Type 8	Type 8	INTERBUS TCP/IP
	Profile 6/3		Type 8	Type 8	Type 8	INTERBUS Subset
		Profile 3/4			Type 8/10	Link 3/4 to 6/1
		Profile 3/5			Type 8/10	Link 3/4 to 6/1
		Profile 3/6			Type 8/10	Link 3/4 to 6/1
Family 7						*Swiftnet* (not in the standard anymore)
Family 8						*CC-Link*
	Profile 8/1		Type 18	Type 18	Type 18	CC-Link/V1
	Profile 8/2		Type 18	Type 18	Type 18	CC-Link/V2
	Profile 8/3		Type 18	Type 18	Type 18	CC-Link/LT (Bus powered-low cost)
		Profile 8/4	8802-3	Type 23	Type 23	CC-Link IE Controller Network
		Profile 8/5	8802-3	Type 23	Type 23	CC-Link IE Field Network
Family 9						*HART*
	Profile 9/1				Type 20	Universal Command (HART 6)
					Type 20	WirelessHART (IEC 62591)
Family 10						*Vnet/IP*
		Profile 10/1	8802-3	UDP/IP	Type 17	Vnet/IP
Family 11						*TCnet*
		Profile 11/1	8802-3	Type 11	Type 11	TCnet
		Profile 11/2	8802-3	Type 11	Type 11	TCnet-loop 100
		Profile 11/3	8802-3	Type 11	Type 11	TCnet-loop 1G
Family 12						*EtherCAT*
		Profile 12/1	Type 12	Type 12	Type 12	Simple IO
		Profile 12/1	Type 12	Type 12	Type 12	Mailbox and time synchronization
Family 13						*Ethernet POWERLINK*
		Profile 13/1	8802-3	Type 13	Type 13	POWERLINK

IEC 61784			IEC 61158 Services and Protocols			
Family	**Part 1**	**Part 2**	**PHL**	**DLL**	**AL**	**Brand Names**
Family 14						*Ethernet for Plant Automation EPA*
		Profile 14/1	8802-3	UDP/TCP/IP	Type 14	EPA Master to Bridge (NRT)
		Profile 14/2	8802-3	Type 14	Type 14	EPA Bridge to Device (RT)
		Profile 14/3	8802-3	Type 14	Type 14	EPA Bridge to Device (FRT)
		Profile 14/4	8802-3	Type 14	Type 14	EPA Bridge to Device (MRT)
Family 15						*MODBUS-RTPS*
		Profile 15/1	8802-3	TCP/IP	Type 15	MODBUS TCP
		Profile 15/2	8802-3	TCP/IP	Type 15	RTPS
Family 16						*SERCOS*
	Profile 16/1		Type 16	Type 16	Type 16	SERCOS I
	Profile 16/2		Type 16	Type 16	Type 16	SERCOS II
		Profile 16/3	8802-3	Type 16	Type 16	SERCOS III
Family 17						*RAPIEnet*
		Profile 17/1	8802-3	Type 21	Type 21	RAPIEnet
Family 18						*SafetyNET p TM*
		Profile 18/1	8802-3	Type 22	Type 22	SafetyNET p RTFL
		Profile 18/2	8802-3	Type 22	Type 22	SafetyNET p RTFN
Family 19						*MECHATROLINK*
	Profile 19/1		Type 24	Type 24	Type 24	MECHATROLINK I
	Profile 19/2		Type 24	Type 24	Type 24	MECHATROLINK II

LIITE B: SFLOW-VIRTAUSNÄYTE MUOKATTUNA ELASTICSEARCH JSON-DOKUMENTIKSI


```
{
  "_index": "sflow-2018.04.17",
  "_type": "doc",
  "_id": "ILSX0mIBe9lZnGcA89ju",
  "_version": 1,
  "_score": null,
  "_source": {
    "node": {
      "ipaddr": "10.155.9.2",
      "hostname": "10.155.9.2"
    },
    "@timestamp": "2018-04-17T07:53:16.381Z",
    "event": {
      "host": "192.168.117.150",
      "type": "sflow"
    },
    "sflow": {
      "dst_hostname": "192.168.117.150",
      "ip_version": "IPv4",
      "service_port": "22",
      "output_interface": "1073741823",
      "dst_port_name": "TCP/22 (ssh)",
      "server_addr": "192.168.117.150",
      "server_hostname": "192.168.117.150",
      "ip_protocol": "TCP",
      "client_addr": "192.168.117.140",
      "vlan": "100",
      "src_port_name": "TCP/58970",
      "tos": "6",
      "dst_port": 22,
      "src_hostname": "192.168.117.140",
      "sampling_interval": 50,
      "dst_addr": "192.168.117.150",
      "src_mac": "00:50:b6:68:b4:90",
      "packets": 50,
      "bytes": 3500,
      "client_hostname": "192.168.117.140",
      "service_name": "TCP/22 (ssh)",
      "input_snmp": "7",
      "src_addr": "192.168.117.140",
      "dst_mac": "70:10:6f:d8:ec:a0",
      "src_port": 58970
      "protocol": "ETHERNET-ISO88023",
      "eth_type": "2048",
      "frame_length": "70",
      "uptime_in_ms": "385500",
      "sub_agent_id": "0",
      "padded": "0",
      "source_id_type": "IF-MIB::ifEntry",
      "sflow_type": "flow_sample",
      "drops": "0",
      "sample_pool": "775",
      "source_id_index": "7",
      "stripped": "0"
    },
  },
  "fields": {
    "@timestamp": [
      "2018-04-17T07:53:16.381Z"
    ]
  },
  "sort": [
    1523951596381
  ]
}
```


Kuva 1. *Logstashin avulla kerätty ja muotoiltu sFlow virtausnäyte muutettuna JSON-dokumentiksi Elasticsearch tallennusta varten.*